BG

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04N 7/16 — A1

(11) International Publication Number: **WO 00/59223**

(43) International Publication Date: 5 October 2000 (05.10.00)

(21) International Application Number: PCT/US00/06473

(22) International Filing Date: 9 March 2000 (09.03.00)

(30) Priority Data:

60/127,178	30 March 1999 (30.03.99)	US
09/422,121	20 October 1999 (20.10.99)	US

(71) Applicant: TIVO, INC. [US/US]; 2160 Gold Street, P.O. Box 2160, Alviso, CA 95002–2160 (US).

(72) Inventors: BARTON, James, M.; 101 Sund Avenue, Los Gatos, CA 95030 (US). BEACH, Brian; 126 Moreno Drive, Santa Cruz, CA 95060 (US).

(74) Agents: GLENN, Michael, A. et al.; Glenn Patent Group, 3475 Edison Way, Suite L, Menlo Park, CA 94025 (US).

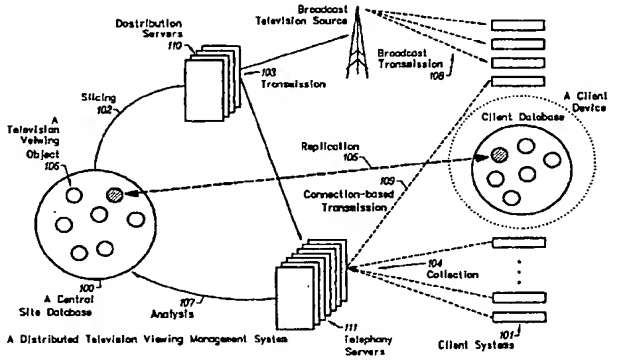
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: DATA STORAGE MANAGEMENT AND SCHEDULING SYSTEM



(57) Abstract

A data storage management and scheduling system schedules the recording, storing, and deleting of television and Web page program material on a client system storage medium. The invention accepts as input a prioritized list of program viewing preferences which is compared with a database of program guide objects which indicate when programs of interest are actually broadcast. A schedule of time versus available storage space is generated that is optimal for the viewer's explicit or derived preferred programs. The preferred programs include television broadcast programs and Universal Resource Locators (URLs). The viewer may request that certain programs be captured, which results in the highest possible priority for those programs, or express preferences using appartenances provided through the viewer interface. Preferences may additionally be inferred from viewing patterns. The invention correlates an input schedule that tracks the free and occupied time slots for each input source with a space schedule that tracks all currently recorded programs and the programs that have been scheduled to be recorded in the future, to schedule new programs to record and resolve recording conflicts. A program is recorded if at all times between when the recording would be initiated and when it expires, sufficient space is available to hold it. All scheduling conflicts are resolved as early as possible. A background scheduler schedules each preferred program in turn until the list of preferred programs is exhausted or no further opportunity to record is available.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

Data Storage Management and Scheduling System

BACKGROUND OF THE INVENTION

TECHNICAL FIELD

The invention relates to the storing and viewing of television program material in a computer environment. More particularly, the invention relates to the management of data on a storage medium in a computer environment.

DESCRIPTION OF THE PRIOR ART

A classic tension exists in the design of automated data processing systems between pure client-server based systems, such as computer mainframe systems or the World Wide Web, and pure distributed systems, such as Networks of Workstations (NOWS) that are used to solve complex computer problems, such as modeling atomic blasts or breaking cryptographic keys.

Client-server systems are popular because they rely on a clean division of responsibility between the server and the client. The server is often costly and specially managed, since it performs computations or stores data for a large number of clients. Each client is inexpensive, having only the local resources needed to interact with the user of the system. A network of reasonable performance is assumed to connect the server and the client. The economic model of these systems is that of centralized management and control driving down the incremental cost of deploying client systems.

However, this model has significant costs that must be considered. For instance, the incremental cost of adding a new client system may be quite high. Additional network capacity must be available, sufficient computing resources must be available to support that client, including storage, memory and computing cycles,

and additional operational overhead is needed for each client because of these additional resources. As the central servers become larger and more complex they become much less reliable. Finally, a system failure of the server results in all clients losing service.

Distributed systems are popular because the resources of the system are distributed to each client, which enables more complex functionality within the client. Access to programs or data is faster since they are located with the client, reducing load on the network itself. The system is more reliable, since the failure of a node affects only it. Many computing tasks are easily broken down into portions that can be independently calculated, and these portions are cheaply distributed among the systems involved. This also reduces network bandwidth requirements and limits the impact of a failed node.

On the other hand, a distributed system is more complex to administer, and it may be more difficult to diagnose and solve hardware or software failures.

Television viewing may be modeled as a client-server system, but one where the server-to-client network path is for all intents and purposes of infinite speed, and where the client-to-server path is incoherent and unmanaged. This is a natural artifact of the broadcast nature of television. The cost of adding another viewer is zero, and the service delivered is the same as that delivered to all other viewers.

There have been, and continue to be, many efforts to deliver television programming over computer networks, such as the Internet, or even over a local cable television plant operating as a network. The point-to-point nature of computer networks makes these efforts unwieldy and expensive, since additional resources are required for each additional viewer. Fully interactive television systems, where the viewer totally controls video streaming bandwidth through a client settop device, have proven even more uneconomical because dedication of server resources to each client quickly limits the size of the system that can be profitably built and managed.

However, television viewers show a high degree of interest in choice and control over television viewing. This interest results in the need for the client system to effectively manage the memory demands of program material that a viewer wants to record. Additionally, the management of recording desired program material is of equal importance to the memory management task.

It would be advantageous to provide a data storage management and scheduling system that manages the available data space on a storage medium and any input sources. It would further be advantageous to provide a data storage management and scheduling system that efficiently schedules the insertion and deletion of data on a medium.

SUMMARY OF THE INVENTION

The invention provides a data storage management and scheduling system. The system schedules the storing and deleting of input source data on a storage medium. In addition, the invention provides a system that manages the available free space on the storage medium such that the available free space is used efficiently.

A client device, typified in Application Serial No. 09/126,071, owned by the Applicant, provides functionality typically associated with central video servers, such as storage of a large amount of video content, ability to choose and play this content on demand, and full "VCR-like" control of the delivery of the content, as typified in Application Serial No. 09/054,604, owned by the applicant.

A preferred embodiment of the invention schedules the recording, storing, and deleting of television and Web page program material on a client system storage medium. The invention accepts as input a prioritized list of program viewing preferences which is compared with a database of program guide objects. The program guide objects indicate when programs of interest are actually broadcast.

A schedule of time versus available storage space is generated that is optimal for the viewer's explicit or derived preferred programs. The preferred programs include television broadcast programs and Universal Resource Locators (URLs). The viewer may request that certain programs be captured, which results in the highest possible priority for those programs.

The viewer may also explicitly express preferences using appurtenances provided through the viewer interface. Preferences may additionally be inferred from viewing patterns. These preferences correspond to objects stored in a replicated database.

The invention correlates an input schedule that tracks the free and occupied time slots for each input source with a space schedule that tracks all currently recorded programs and the programs that have been scheduled to be recorded in the future, to schedule new programs to record and resolve recording conflicts. A program is recorded if at all times between when the recording would be initiated and when it expires, sufficient space is available to hold it. Programs scheduled for recording based on inferred preferences automatically lose all conflict decisions. All scheduling conflicts are resolved as early as possible. Schedule conflicts resulting from the recording of aggregate objects are resolved using the preference weighting of the programs involved.

A background scheduler attempts to schedule each preferred program in turn until the list of preferred programs is exhausted or no further opportunity to record is available. A preferred program is scheduled if and only if there are no conflicts with other scheduled programs

Other aspects and advantages of the invention will become apparent from the following detailed description in combination with the accompanying drawings, illustrating, by way of example, the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block schematic diagram of a preferred embodiment of a distributed television viewing management system according to the invention;

Fig. 2 is a block schematic diagram of the structure of a viewing object in computer storage for programmatic access according to the invention;

Fig. 3 is a block schematic diagram showing how the schema for a viewing object is structured in computer storage for programmatic access according to the invention;

Fig. 4 is a block schematic diagram showing an example graph of relationships between viewing objects which describe information about programs according to the invention;

Fig. 5 is a block schematic diagram showing an example graph of relationships generated when processing viewer preferences to determine programs of interest according to the invention;

Fig. 6 is a block schematic diagram showing the scheduling of inputs and storage space for making recordings according to the invention;

Fig. 7 is a flowchart showing the steps taken to schedule a recording using the mechanism depicted in Fig. 6 according to the invention;

Fig. 8 is a block schematic diagram of a preferred embodiment of the invention showing the bootstrap system configuration according to the invention;

Fig. 9a is a block schematic diagram of the decision flowchart for the bootstrap component according to the invention;

Fig. 9b is a block schematic diagram of the decision flowchart for the bootstrap component according to the invention; and

Fig. 10 is a block schematic diagram of the decision flowchart for the software installation procedure according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

The invention is embodied in a data storage management and scheduling system in a computer environment. A system according to the invention schedules the storing and deleting of input source data on a storage medium. In addition, the invention provides a system that manages the available free space on the storage medium such that the available free space is used efficiently.

The invention is exemplified as part of a television viewing information transmission and collection system that improves the ability of the individual viewer to select and automatically timeshift television programs while providing opportunities for a service provider to enhance and direct the viewing experience. The following describes a system which is fully distributed, in that calculations pertaining to an individual viewer are performed personally for that viewer within a local client device, while providing for the reliable aggregation and dissemination of information concerning viewing habits, preferences or purchases.

The Database of Television Viewing Information

Fig. 1 gives a schematic overview of the invention. Central to the invention is a method and apparatus for maintaining a distributed database of television viewing information among computer systems at a central site 100 and an extremely large number of client computing systems 101. The process of extracting suitable subsets of the central copy of the database is called "slicing" 102, delivering the resulting "slices" to clients is called "transmission" 103, delivering information collected about or on behalf of the viewer to the central site is called "collection" 104, and processing the collected information to generate new television viewing objects or reports is called "analysis" 107; in all cases, the act of recreating an object from one database within another is called "replication" 105. Data items to be transmitted or collected are termed "objects" 106, and the central database and each replicated subset of the central database contained within a client device is an "object-based" database. The objects within this database are often termed "television viewing objects", "viewing objects", or simply "objects", emphasizing their intended use. However, one skilled in the art will readily appreciate that objects can be any type of data.

The viewing object database provides a consistent abstract software access model for the objects it contains, independent of and in parallel with the replication activities described herein. By using this interface, applications may create, destroy, read, write and otherwise manipulate objects in the database without concern for underlying activities and with assurance that a consistent and reliable view of the objects in the database and the relationships between them is always maintained.

Basic Television Viewing Object Principles

Referring to Fig. 2, television viewing objects are structured as a collection of "attributes" 200. Each attribute has a type 201, *e.g.*, integer, string or boolean, and a value 202. All attribute types are drawn from a fixed pool of basic types supported by the database.

The attributes of an object fall into two groups: "basic" attributes, which are supplied by the creator or maintainer of the viewing object; and "derived" attributes, which are automatically created and maintained by mechanisms within the database. Basic attributes describe properties of the object itself; derived attributes describe the relationships between objects. Basic attributes are replicated between databases, whereas derived attributes are not.

With respect to Fig. 3, there is a small set of fundamental object types defined by the invention; each object type is represented as a specific set of related attributes 300, herein called a "schema". The schema defines a template for each attribute type 301, which includes the type 302 and name of the attribute 303. Actual television viewing objects are created by allocating resources for the object and assigning values to the attributes defined by the schema. For example, a "program" schema might include attributes such as the producer, director or actors in the program, an on-screen icon, a multi-line description of the program contents, an editorial rating of the program, etc. A physical program object is created by allocating storage for it, and filling in the attributes with relevant data.

There is one special object type predefined for all databases called the schema type. Each schema supported by the database is represented by a schema object. This allows an application to perform "introspection" on the database, *i.e.*, to dynamically discover what object types are supported and their schema. This greatly simplifies application software and avoids the need to change application

software when schemas are changed, added or deleted. Schema objects are handled the same as all other viewing objects under the methods of this invention.

Referring again to Fig. 2, each object in a database is assigned an "object ID" 203 which must be unique within the database. This object ID may take many forms, as long as each object ID is unique. The preferred embodiment uses a 32-bit integer for the object ID, as it provides a useful tradeoff between processing speed and number of unique objects allowed. Each object also includes a "reference count" 204, which is an integer giving the number of other objects in the database which refer to the current object. An object with a reference count of zero will not persist in the database (see below).

One specific type of viewing object is the "directory" object. A directory object maintains a list of object IDs and an associated simple name for the object. Directory objects may include other directory objects as part of the list, and there is a single distinguished object called the "root" directory. The sequence of directory objects traversed starting at the root directory and continuing until the object of interest is found is called a "path" to the object; the path thus indicates a particular location within the hierarchical namespace created among all directory objects present in the database. An object may be referred to by multiple paths, meaning that one object may have many names. The reference count on a viewing object is incremented by one for each directory which refers to it.

Methods for the Maintenance of Database Consistency and Accuracy

One of the features of a preferred embodiment of the invention is to insure that each database replica remains internally consistent at all times, and that this consistency is automatically maintained without reference to other databases or the need for connection to the central site. There is no assurance that transmission or collection operations happen in a timely manner or with any assured periodicity. For instance, a client system may be shut off for many months; when a transmission to the system is finally possible, the replication of objects must always result in a consistent subset of the server database, even if it is not possible to transmit all objects needed to bring the central and client databases into complete synchronization.

Even more serious, there can be no guarantee of a stable operational environment while the database is in use or being updated. For example,

electrical power to the device may cease. This invention treats all database updates as "transactions", meaning that the entire transaction will be completed, or none of it will be completed. The specific technique chosen is called "two-phase commit", wherein all elements of the transaction are examined and logged, followed by performing the actual update. One familiar in the art will appreciate that a standard journaling technique, where the transaction is staged to a separate log, combined with a roll-forward technique which uses the log to repeat partial updates that were in progress when the failure occurred, is sufficient for this purpose.

One required derived attribute of every object is the "version", which changes with each change to the object; the version attribute may be represented as a monotonically increasing integer or other representation that creates a monotonic ordering of versions. The schema for each object that may be replicated includes an attribute called "source version" which indicates the version of the object from which this one was replicated.

Transmission of a viewing object does not guarantee that every client receives that object. For instance, while the object is being broadcast, external factors such as sunspots, may destroy portions of the transmission sequence. Viewing objects may be continually retransmitted to overcome these problems, meaning that the same object may be presented for replication multiple times. It is inappropriate to simply update the database object each time an object to be replicated is received, as the version number will be incremented although no change has actually occurred. Additionally, it is desirable to avoid initiating a transaction to update an object if it is unnecessary; considerable system resources are consumed during a transaction.

Two approaches are combined to resolve this problem. First, most objects will have a basic attribute called "expiration". This is a date and time past which the object is no longer valid, and should be discarded. When a new object is received, the expiration time is checked, and the object discarded if it has expired. Expiration handles objects whose transmission is delayed in some fashion, but it does not handle multiple receptions of the same unexpired object.

The source version attribute handles this problem. When a viewing object is transmitted, this attribute is copied from the current version attribute of the source object. When the viewing object is received, the source version of the received object is compared with the source version of the current object. If the new object

has a higher source version attribute, it is copied over the existing object, otherwise it is discarded.

It is assumed that a much greater number of viewing objects are transmitted than are of interest to any particular client system. For example, a "channel" viewing object which describes the channels on a particular cable system is of no interest to clients attached to other cable systems. Because of the overhead of capturing and adding new objects to the database, it would be advantageous for received objects to be filtered on other attributes in addition to those described above. The invention accomplishes this by using a filtering process based on object type and attribute values. In one implementation, this filtering process is based on running executable code of some kind, perhaps as a sequence of commands, which has been written with specific knowledge of various object types and how they should be filtered.

In a preferred embodiment of the invention, a "filter" object is defined for each object type which indicates what attributes are required, should not be present, or ranges of values for attributes that make it acceptable for addition to the database. One skilled in the art will readily appreciate that this filter object may contain executable code in some form, perhaps as a sequence of executable commands. These commands would examine and compare attributes and attribute values of object being filtered, resulting in an indication of whether the object should be the subject of further processing.

Viewing objects are rarely independent of other objects. For example, a "showing" object (describing a specific time on a specific channel) is dependent on a "program" object (describing a specific TV program). One important aspect of maintaining consistency is to insure that all dependent objects either already exist in the database or are to be added as part of a single transaction before attempting to add a new viewing object. This is accomplished using a basic attribute of the new viewing object called the "dependency" attribute, which simply lists the object IDs and source versions of objects that the new object is dependent on. Clearly, new versions of an object must be compatible, in the sense that the schema defining new versions be the same or have a strict superset of the attributes of the original schema.

When a new viewing object is received, the database is first checked to see if all dependencies of that object are present; if so, the object is added to the database. Otherwise, the new object is "staged", saving it in a holding area until

all dependent objects are also staged. Clearly, in order for a new set of viewing objects to be added to the database, the dependency graph must be closed between objects in the staging area and objects already existing in the database, based on both object ID and source version. Once closure is achieved, meaning all dependent objects are present, the new object(s) are added to the database in a single atomic transaction.

Naming and Finding Television Viewing Objects

Directory objects have been described previously. Referring to Fig. 4, the collection of directory objects, and the directed graph formed by starting at the root path 400 and enumerating all possible paths to viewing objects is called a "namespace". In order for an object to be found without knowing a specific object ID, one or more paths within this namespace must refer to it. For instance, application software has little interest in object IDs, instead the software would like to refer to objects by paths, for instance "/tvschedule/today". In this example, the actual object referred to may change every day, without requiring changes in any other part of the system.

One way in which a path to an object may be established is by specifying a "pathname" basic attribute on the object. The object is added to the database, and directory objects describing the components of the path are created or updated to add the object. Such naming is typically used only for debugging the replication mechanisms. Setting explicit paths is discouraged, since the portions of the central database replicated on each client system will be different, leading to great difficulty in managing pathnames among all replicas of the database.

A preferred method for adding an object to the database namespace is called "indexing". In a preferred embodiment of the invention, an "indexer" object is defined for each object type which indicates what attributes are to be used when indexing it into the database namespace. One skilled in the art will readily appreciate that this indexer object may contain executable code in some form, perhaps as a sequence of executable commands. These commands would examine and compare attributes and attribute values of object being indexed, resulting in an indication of where the object should be located in the namespace.

Based on the object type, the indexer examines a specific set of attributes attached to the object. When such attributes are discovered the indexer automatically adds a name for the object, based on the value of the attribute,

within the hierarchical namespace represented by the graph of directories in the database. Referring again to Fig. 4, a program object may have both an "actor" attribute with value "John Wayne" and a "director" attribute with value "John Ford" 401. The root directory might indicate two sub-directories, "byactor" 402 and "bydirector" 403. The indexer would then add the paths "/byactor/John Wayne" and "/bydirector/John Ford" to the database, both of which refer to the same object 401.

A derived attribute is maintained for each object listing the directory objects which refer to this object 404. As the indexer adds paths to the namespace for this object, it adds the final directory ID in the path to this list. This insures closure of the object graph – once the object has been found, all references to that object within the database are also found, whether they are paths or dependencies.

This unique and novel method of adding objects to the database has significant advantages over standard approaches. The indexer sorts the object into the database when it is added. Thus, the search for the object associated with a particular path is a sequence of selections from ordered lists, which can be efficiently implemented by one familiar with the art.

Deleting Objects from the Database

While the rules for adding objects to the database are important, the rules for removing objects from the database are also important in maintaining consistency and accuracy. For example, if there were no robust rules for removing objects, the database might grow unboundedly over time as obsolete objects accumulate.

The cardinal rule for deleting objects from the database is based on reference counting; an object whose reference count drops to zero is summarily deleted. For instance, this means that an object must either be referred to by a directory or some other object to persist in the database. This rule is applied to all objects in the closed dependency graph based on the object being deleted. Thus, if an object which refers to other objects (such as a directory) is deleted, then the reference count on all objects referred to is decremented, and those objects similarly deleted on a zero count, and so forth.

There is also an automatic process which deletes objects from the database called the "reaper". Periodically, the reaper examines all objects in the database, and depending on the object type, further examines various attributes and attribute values to decide if the object should be retained in the database. For example, the expiration attribute may indicate that the object is no longer valid, and the reaper will delete the object.

In the preferred embodiment, using a method similar to (or perhaps identical to) the filtering and indexing methods described above, the reaper may instead access a reaper object associated with the object type of the current object, which may contain executable code of various kinds, perhaps a sequence of executable commands. This code examines the attributes and attribute values of the current object, and determines if the object should be deleted.

The overhead of individually deleting every object for which the reference count has been decremented to zero may be quite high, since every such deletion results in a transaction with the database. It would be advantageous to limit the performance impact of reaping objects, such that foreground operations proceed with maximum speed. In a preferred embodiment, this is accomplished using a technique based on common garbage collection methods.

For instance, instead of deleting an object whose reference count has been decremented to zero, the reaper performs no other action. Periodically, a background task called the garbage collector examines each object in the database. If the object has a reference count of zero, it is added to a list of objects to be deleted. In one embodiment, once the garbage collector has examined the entire database, it would delete all such objects in a single transaction. One familiar in the art will appreciate that this method may also result in a significant performance penalty, as other accesses to the database may be delayed while the objects are being deleted. In addition, if all objects are to be properly deleted, changes to the database may have to be delayed while the garbage collector is active, resulting in even worse performance.

In a preferred embodiment, the garbage collector examines the database in a series of passes. Once a specific number of objects has been collected, they are deleted in a single transaction. Said process continues until all objects have been examined. This technique does not guarantee that all garbage objects are collected during the examination process, since parallel activities may release objects previously examined. These objects will be found, however, the next

time the garbage collector runs. The number of objects deleted in each pass is adjustable to achieve acceptable performance for other database activities.

Operations on the Distributed Television Viewing Object Database

Considerations in Maintaining the Distributed Viewing Object Database

The replication of television viewing objects among the instances of the distributed database necessarily requires the transmission of objects over unreliable and unsecure distribution channels.

For example, if the objects are transmitted over a broadcast mechanism, such as within a radio or television transmission, there can be no assurance that the data is transmitted accurately or completely. Weather, such as rainstorms, may cause dropouts in the transmission. Other sources of interference may be other broadcast signals, heavy equipment, household appliances, etc.

One skilled in the art will readily appreciate that there are standard techniques for managing the transmission of data over unreliable channels, including repeated transmissions, error correcting codes, and others, which may be used for transmission, any or all of which may be used in any particular instance.

For efficiency, objects to be replicated are gathered together into distribution packages, herein called "slices". A slice is a subset of the television viewing object database which is relevant to clients within a specific domain, such as a geographic region, or under the footprint of a satellite transmitter.

Security of these slices is quite important. Slices are used to add objects to the database which are used to provide valuable services to users of the database, as well as to store information that may be considered private or secret. Because of the broadcast-oriented nature of slice transmission, slices may be easily copied by third parties as they are transmitted. A practical solution to these problems is to encrypt the slice during transmission. An ideal reference text on the techniques employed in the invention is "Applied Cryptography: Protocols, Algorithms, and Source Code in C" by Bruce Schneier, John Wiley and Sons, 1995.

In a preferred embodiment of the invention, a secure, encrypted channel is established using techniques similar to those described in U.S. Pat. Serial No. 4,405,829, often described as asymmetric key encryption, or sometimes public/private key pair encryption. A practitioner skilled in the art will recognize that protocols based on asymmetric key encryption serves as a reliable and efficient foundation for authentication of client devices and secure distribution of information. In general, authentication is provided using an exchange of signed messages between the client and central systems. Secure distribution is provided by encrypting all communications using a short-lived symmetric key sent during an authentication phase.

Successful security requires that sender and receiver agree beforehand on the asymmetric key pair to be used for encryption. Such key distribution is the weakest link in any cryptographic system for protecting electronic data. Application Serial No. 09/357,183, entitled "Self-Test Electronic Assembly and Test System," filed July 19, 1999, also owned by the Applicant, describes a mechanism whereby the client device generates the asymmetric key pair automatically as the final step in the manufacturing process. The private key thus generated is stored within a secure microprocessor embedded within the client device, such that the key is never presented to external devices. The public key thus generated is transmitted to a local manufacturing system, which records the key along with the client serial number in a secure database. This database is later securely transmitted to the central distribution system, where it is used to perform secure communications with the client.

This unique and novel application of key generation solves the problem of key distribution, as the private key is never presented to external components in the client, where it might be discerned using special tools, such as a logic analyzer. Instead, it may only be used within the security microprocessor itself to decrypt messages originally encrypted with the public key, the results of which are then provided to external components.

The remainder of this discussion assumes that all communications between client and central systems are authenticated and encrypted as described above.

Transmitting Viewing Objects to the Client Systems

Referring again to Fig. 1, in a preferred embodiment of the invention the following steps constitute "transmission" of television viewing objects from the central database using slices:

1. There may be many mechanisms for transmitting slices to the universe of client viewing devices. For instance, the slices may be directly downloaded over a telephone modem or cable modem 109, they may be modulated into lines of the Vertical Blanking Interval (VBI) of a standard television broadcast 108 , or added to a digital television multiplex signal as a private data channel. One skilled in the art will readily appreciate that any mechanism which can transmit digital information may be used to transmit slices of the television viewing object database.

The first step in preparing television viewing objects for transmission is recognizing the transmission mechanism to be used for this particular instance, and creating a slice of a subset of the database that is customized for that mechanism. For example, the database may contain television viewing objects relating to all programs in the country. However, if television viewing objects are to be sent using VBI modulation on a local television signal, only those television viewing objects relating to programs viewable within the footprint of the television broadcast being used to carry them should be contained within the relevant slice. Alternatively, if some of the television viewing objects contain promotional material related to a particular geographic region, those objects should not be transmitted to other geographic regions.

In a preferred embodiment of the invention, the speed and periodicity of traversing the database and generating slices for transmission is adjustable in an arbitrary fashion to allow useful cost/performance tradeoffs to be made. For instance, it may only be necessary to create slices for certain transmission methods every other day, or every hour.

The final step in preparing each slice is to encrypt the slice using a short-lived symmetric key. Only client devices which have been authenticated using secure protocols will have a copy of this symmetric key, making them able to decrypt the slice and access the television viewing objects within it.

2. Once a slice is complete, it is copied to the point at which the transmission mechanism can take and send the data 110. For telephone connections, the slice is placed on a telephony server 111 which provides the data to each client as it calls in. If television broadcast is used, the slice is copied onto equipment co-resident with the station television transmitter, from whence it is modulated onto the signal. In these and similar broadcast-oriented cases, the slice is "carouseled", *i.e.*, the data describing the slice is repeated continually until a new slice is provided for transmission.

   This repetitive broadcast of slices is required because there can be no assurance that the signal carrying the data arrives reliably at each client. The client device may be powered off, or there may be interference with reception of the signal. In order to achieve a high degree of probability that the transmitted slices are properly received at all client devices, they are continually re-broadcast until updated slices are available for transmission.

   A preferred embodiment of the invention uses broadcast mechanisms such as a television signal to transmit the slice. However, it is desirable to provide for download over a connection-based mechanism, such as a modem or Internet connection. Using a connection-based mechanism usually results in time-based usage fees, making it desirable to minimize the time spent transmitting the slice.

   This is accomplished using a two-step process. When the connection is established, the client system sends an inventory of previously received slices to telephony servers 111. The server compares this inventory with the list of slices that should have been processed by that client. Slices which were not processed are transmitted to the client system.

3. The slice is transmitted by breaking the encrypted slice into a succession of short numbered data packets. These packets are captured by client systems and held in a staging area until all packets in the sequence are present. The packets are reassembled into the slice, which is then decrypted. The television viewing objects within the slice are then filtered for applicability, possibly being added to the local television viewing object database. This process replicates a portion of the central database of television viewing objects reliably into the client.

The invention keeps track of the time at which data packets are received. Data packets which are older than a selected time period are purged from the staging area on a periodic basis; this avoids consuming space for an indefinite period while waiting for all parts of a slice to be transmitted.

Especially when transmitting the objects over a broadcast medium, errors of various kinds may occur in the transmitted data. Each data packet is stamped with an error detecting code (a parity field or CRC code, for example). When an error is detected the data packet is simply discarded. The broadcast carousel will eventually retransmit the data packet, which is likely to be received properly. Slices of any size may thus be sent reliably; this is achieved at the cost of staging received portions of the object on the client until all portions are properly received.

4. There may be one or more "special" slices transmitted which communicate service related data to the client system, particularly service authorization information. It is important that the service provider be able to control the client system's access to premium services if the viewer has failed to pay his bill or for other operational reasons.

One particular type of special slice contains an "authorization" object. Authorization objects are generally encrypted using asymmetric key encryption based on the public/private key pair associated with a specific client. If the slice can be successfully decrypted by the security microprocessor using the embedded private key, the slice will contain an object indicating the allowable time delay before another authorization object is received, as well as one or more symmetric keys valid for a short time period. The delay value is used to reset a timestamp in the database indicating when the client system will stop providing services. The symmetric keys are stored in the local television viewing object database, to be used in decrypting new slices which may be received.

If the client has not received a proper authentication object by the time set in the database, it will commence denial of most services to the viewer (as specified by the service provider). Also contained within an authentication object are one or more limited-lifetime download keys which are needed to decrypt the slices that are transmitted. Clearly, if a client system is unable to authenticate itself, it will not be able to decrypt any objects.

Each authorization slice is individually generated and transmitted. If broadcast transmission is used for the slices, all relevant authorizations are treated identically to all other slices and carouseled along with all other data. If direct transmission is used, such as via a phone connection, only the authentication slice for that client is transmitted.

5. Once the client device has received a complete database slice, it uses the methods described earlier to add the new object contained within it to the database.

Collecting Information from the Client Systems

Referring again to Fig. 1, in a preferred embodiment of the invention the following steps constitute "collection" of television viewing objects from each client database:

1. As the viewer navigates the television channels available to him, the client system records interesting information, such as channel tuned to, time of tuning, duration of stay, VCR-like actions (*e.g.*, pause, rewind), and other interesting information. This data is stored in a local television viewing object.

   Additionally, the viewer may indicate interest in offers or promotions that are made available, or he may indicate a desire to purchase an item. This information is also recorded into a local television viewing object.

   Additionally, operation of the client device may result in important data that should be recorded into a television viewing object. For example, errors may occur when reading from the hard disk drive in the client, or the internal temperature of the device may exceed operational parameters. Other similar types of information might be failure to properly download an object, running out of space for various disk-based operations, or rapid power cycling.

2. At a certain time, which may be immediate or on a periodic basis, the client system contacts the central site via a direct connection 104 (normally via phone and/or an Internet connection). The client device sends a byte

sequence identifying itself which is encrypted with its secret key. The server fetches the matching television viewing object for the client device from the database, and uses the key stored there to decrypt the byte sequence. At the same time, the server sends a byte sequence to the client, encrypted in its secret key, giving the client a new one-time encryption key for the session.

Both sides must successfully decrypt their authentication message in order to communicate. This two-way handshake is important, since it assures both client and server that the other is valid. Such authentication is necessary to avoid various attacks that may occur on the client system. For example, if communications were not authenticated in such a fashion, a malicious party might create an "alias" central site with a corrupt television viewing object database and provide bad information to a client system, causing improper operation. All further communication is encrypted using the one-time session key. Encrypted communication is necessary because the information may pass across a network, such as the Internet, where data traffic is open to inspection by all equipment it passes through. Viewing objects being collected may contain information that is considered private, so this information must be fully protected at all times.

Assuming that the authentication phase is successful, the two parties treat the full-duplex phone line as two one-way broadcast channels. New slices are delivered to the client, and viewing data to be collected is sent back. The connection is ended when all data is delivered.

One skilled in the art will readily appreciate that this connection may take place over a network, such as the Internet running standard TCP/IP protocols, transparently to all other software in the system.

3. Uploaded information is handled similarly by the server; it is assumed to represent television viewing objects to be replicated into the central database. However, there may be many uploaded viewing objects, as there may be many clients of the service. Uploaded objects are therefore assigned a navigable attribute containing information about their source; the object is then indexed uniquely into the database namespace when it is added.

Uploaded viewing objects are not immediately added to the central database; instead they are queued for later insertion into the database. This

step allows the processing of the queue to be independent of the connection pattern of client devices. For instance, many devices may connect at once, generating a large number of objects. If these objects were immediately added to the central database, the performance of all connections would suffer, and the connection time would increase. Phone calls are charged by duration, thus any system in which connection time increases as a function of load is not acceptable.

Another advantage of this separation is that machine or network failures are easily tolerated. In addition, the speed at which viewing objects are processed and added to the central database may be controlled by the service provider by varying the computer systems and their configurations to meet cost or performance goals.

Yet another advantage of this separation is that it provides a mechanism for separating data collected to improve service operations and data which might identify an individual viewer. It is important that such identifying data be kept private, both for legal reasons and to increase the trust individuals have in the service. For instance, the navigable attribute assigned to a viewing object containing the record of a viewer's viewing choices may contain only the viewer's zip code, meaning that further processing of those objects can construct no path back to the individual identity.

Periodic tasks are invoked on the server to cull these objects from the database and dispose of them as appropriate. For example, objects indicating viewer behavior are aggregated into an overall viewer behavior model, and information that might identify an individual viewer is discarded. Objects containing operational information are forwarded to an analysis task, which may cause customer service personnel to be alerted to potential problems. Objects containing transactional information are forwarded to transaction or commerce systems for fulfillment.

Any of these activities may result in new television viewing objects being added to the central database, or in existing objects being updated. These objects will eventually be transmitted to client devices. Thus, the television viewing management system is closed loop, creating a self-maintaining replicated database system 105 which can support any number of client systems.

Processing of Television Viewing Objects by Client Systems

Television viewing objects may contain the following types of information: television program descriptions and showing times; cable, satellite or broadcast signal originator information, such as channel numbering and identification; viewer preference information, such as actors, genre, showing times, etc.; software, such as enhanced database software, application software, operating system software, etc.; statistical modeling information such as preference vectors, demographic analysis, etc.; and any other arbitrary information that may be represented as digital data.

Methods Applied to Program Guide Objects

Program guide objects contain all information necessary for software running in the client system to tune, receive, record and view programs of interest to the user of the client system, selecting from among all available programs and channels as described by objects within the database.

This program guide information is updated on a regular basis by a service provider. This is handled by the provider acquiring program guide information in some manner, for instance, from a commercial supplier of such information or other sources of broadcast schedule information. This data is then processed using well-understood software techniques to reduce the information to a collection of inter-related viewing objects.

Referring again to Fig. 4, a typical relationship between program guide objects is shown. A television "network" object 407 is any entity which schedules and broadcasts television programming, whether that broadcast occurs over the air, cable, satellite, or other suitable medium. A television "program" object 401 is a description of any distinct segment of a television broadcast signal, such as a particular program, commercial advertisement, station promotion, opener, trailer, or any other bounded portion of a television signal. A "showing" object 406 is a portion of the broadcast schedule for a network on which a program is broadcast. A "channel map" object maps a network broadcast onto a particular broadcast

channel for the medium being used; for instance, a channel map object for a satellite broadcast service would include information about the transponder and data stream containing the broadcast. Using the previously described methods, this program guide data is replicated from the central site to the client systems, where application software in the client systems use the data to manage television viewing.

The service provider may also provide aggregation viewing objects, which describe a set of program guide objects that are interrelated in some fashion. For instance, a "Star-Trek" collection might contain references to all program guide objects associated with this brand name. Clearly, any arbitrary set of programs may be aggregated in this fashion. Aggregation objects are similar to directories. For instance, the Star Trek collection might be found at "/showcases/Star Trek" in the hierarchical namespace. Aggregation objects are also program guide objects, and may be manipulated in a similar fashion, including aggregating aggregation objects, and so forth.

The client system may further refine the collection of program objects. In a system where programming may be captured to internal storage, each captured program is represented by a new program guide object, becoming available for viewing, aggregation, etc. Explicit viewer actions may also result in creation of program guide objects. For instance, the viewer may select several programs and cause creation of a new aggregation object.

This description of types of program guide objects is not meant to be inclusive; there may be many different uses and ways of generating program guide objects not herein described which still benefit from the fundamental methods of the invention.

Program guide objects are used by the application software in five ways:

1. In the simplest case, the viewer may wish to browse these objects to discern current or soon-to-be-available programming. The application software will map the object relationships described by the database to some form of visual and audible interface that is convenient and useful for the viewer. The

viewer may indicate that a particular program is of interest, resulting in some application-specific action, such as recording the program to local storage when it is broadcast.

2. Application software may also directly process program guide objects to choose programs that may be of interest to the viewer. This process is typically based on an analysis of previously watched programming combined with statistical models, resulting in a priority ordering of all programs available. The highest priority programs may be processed in an application specific manner, such as recording the program to local storage when it is broadcast. Portions of the priority ordering so developed may be presented to the viewer for additional selection as in case 1.

One skilled in the art will readily appreciate that there is a great deal of prior art centered on methods for selecting programming for a viewer based on previous viewing history and explicit preferences, *e.g.*, U.S. Pat. Serial No. 5,758,257. The methods described in this application are unique and novel over these techniques as they suggest priorities for the capture of programming, not the broadcast or transmission of programming, and there is no time constraint on when the programming may be broadcast. Further details on these methods are given later in this description.

In general, explicit viewer choices of programming have the highest priority for capture, followed by programming chosen using the preference techniques described herein.

3. A client system will have a small number of inputs capable of receiving television broadcasts or accessing Web pages across a network such as an intranet or the Internet. A scheduling method is used to choose how each input is tuned, and what is done with the resulting captured television signal or Web page.

Referring to Fig. 6, generally, the programs of interest to the viewer may be broadcast at any time, on any channel, as described by the program guide objects. Additionally, the programs of interest may be Web page Universal Resource Locators (URL) across a network, such as an intranet or the Internet.

The channel metaphor is used to also describe the location, or URL, of a particular Web site or page.

A viewer, for example, can "tune" into a Web site by designating the Web site URL as a channel. Whenever that channel is selected, the Web site is displayed. A Web page may also be designated as a program of interest and a snapshot of the Web page will be taken and recorded at a predetermined time.

The scheduler accepts as input a prioritized list of program viewing preferences 603, possibly generated as per the cases above. The scheduling method 601 then compares this list with the database of program guide objects 604, which indicate when programs of interest are actually broadcast. It then generates a schedule of time 607 versus available storage space 606 that is optimal for the viewer's explicit or derived preferred programs. Further details on these methods are given later in this description.

4. When a captured program is viewed, the matching program guide object is used to provide additional information about the program, overlaid on the display using any suitable technique, preferably an On Screen Display (OSD) of some form. Such information may include, but is not limited to: program name; time, channel or network of original broadcast; expiration time; running time or other information.

5. When live programming is viewed, the application uses the current time, channel, and channel map to find the matching program guide object. Information from this object is displayed using any suitable technique as described above. The information may be displayed automatically when the viewer changes channels, when a new program begins, on resumption of the program after a commercial break, on demand by the viewer, or based on other conditions.

6. Using techniques similar to those described in case 2, application software may also capture promotional material that may be of interest to the viewer. This information may be presented on viewer demand, or it may be

automatically inserted into the output television signal at some convenient point. For example. an advertisement in the broadcast program might be replaced by a different advertisement which has a higher preference priority. Using the time-warping apparatus, such as that described in Application Serial No. 09/126,071, entitled "Multimedia Time Warping System," filed July 30, 1998, it is possible to insert any stored program into the output television signal at any point. The time-warping apparatus allows the overlaid program to be delayed while the stored program is inserted to make this work.

Methods for Generating a List of Preferred Programs

Viewer preferences may be obtained in a number of ways. The viewer may request that certain programs be captured, which results in the highest possible priority for those programs. Alternatively, the viewer may explicitly express preferences using appurtenances provided through the viewer interface, perhaps in response to a promotional spot for a particular program, or even during the viewing of a program. Finally, preferences may be inferred from viewing patterns: programs watched, commercial advertisements viewed or skipped, etc.

In each case, such preferences must correspond to television viewing objects stored in the replicated database. Program objects included a wealth of information about each particular program, for example: title, description, director, producer, actors, rating, etc. These elements are stored as attributes attached to a program object.

Each individual attribute may result in the generation of a preference object. Such objects store the following information:

1. The type of the preference item, such as actor or director preference;
2. The weight of the preference given by the viewer, which might be indicated by multiple button presses or other means;

3. The statically assigned significance of the preference in relation to other preferences, for example, actor preference are more significant than director preferences;

4. The actual value of the preference item, for instance the name of the director.

With respect to Fig. 5, preference objects are stored in the database as a hierarchy similar to that described for program guide objects, however this hierarchy is built incrementally as preferences are expressed 500. The hierarchy thus constructed is based on "direct" preferences, *e.g.*, those derived from viewer actions or inferred preferences.

A similar hierarchy is developed based on "indirect" preferences pointing to the same preference objects 501. In general, indirect preferences are generated when preferences for aggregate objects are generated, and are used to further weight the direct preferences implied by the collection of aggregated objects. The preference objects referenced through the indirect preference hierarchy are generated or updated by enumerating the available program objects which are part of the aggregate object 502, and generating or updating preference objects for each attribute thus found.

The weight of a particular preference 503 begins at zero, and then a standard value is added based on the degree of preference expressed (perhaps by multiple button presses) or a standard value is subtracted if disinterest has been expressed. If a preference is expressed based on an aggregate viewing object, all preferences generated by all viewing objects subordinate to the aggregated object are similarly weighted. Therefore, a new weighting of relevant preference elements is generated from the previous weighting. This process is bounded by the degree of preference which is allowed to be expressed, thus all weightings fall into a bounded range.

In a preferred embodiment of the invention, non-linear combinations may be used for weighting a preference item. For instance, using statistical models provided by the central site, the client may infer that a heavily weighted preference for three attributes in conjunction indicates that a fourth attribute should be heavily weighted as well.

The list of preferred programs is generated as follows:

1. A table 504 is constructed which lists each possible program object attribute, and any preference objects for that attribute that are present are listed in that entry.
2. If the preference item is a string, such as an actor name, a 32-bit digital signature for that string is calculated using a 32-bit CRC algorithm and stored with the table item, rather than the string itself. This allows for much faster scanning of the table as string comparisons are avoided, at the slight risk of two different strings generating the same digital signature.
3. For each program object in the database, and for each attribute of that program, the attribute is looked up in the table. If present, the list of preference objects for that attribute is examined for a match with the attribute of the current program object. If a match occurs, the weight associated with that preference object is added to weighting associated with the program object to generate a single weight for the program.
4. Finally, the program objects are rank-ordered based on the overall weighting for each program, resulting in a list of most-preferred to least-preferred programs.

Given this final prioritized list, a recording schedule is generated using the methods described below, resulting in a collection of recorded programs of most interest to the viewer.

Methods applied to scheduling recording versus available storage space

As has been described previously, recorded programs will in general have an expiration date, after which the recorded program is removed from client storage. The viewer may at any time indicate that a program should be saved longer, which delays expiration by a viewer-selected interval. The invention views the available storage for recording programs as a "cache"; unviewed programs are removed after a time, based on the assumption they will not be watched if not watched soon after recording. Viewed programs become immediate candidates for deletion, on the assumption they are no longer interesting.

With proper scheduling of recording and deletion of old programs, it is possible to make a smaller storage area appear to be much larger, as there is an ongoing flushing of old programs and addition of new programs. Additionally, if resources are available, recordings may be scheduled of programs based on inferred preferences of the viewer; these are called "fuzzy" recordings. This results in a system where the program storage area is always "full" of programming of interest to the viewer; no program is removed until another program is recorded in its place or the viewer explicitly deletes it.

Additionally, the viewer may select a program for recording at any time, and the recording window may conflict with other scheduled recordings, or there may not be sufficient space obtainable when the program must be recorded. The invention includes unique and novel methods of resolving such conflicts.

Conflicts can arise for two reasons: lack of storage space, or lack of input sources. The television viewing system described herein includes a fixed number of input sources for recording video and a storage medium, such as a magnetic disk, of finite capacity for storing the recorded video. Recording all television programs broadcast over any significant period of time is not possible. Therefore, resolving the conflicts that arise because of resource limitations is the key to having the correct programs available for viewing.

Referring again to Fig 6, the invention maintains two schedules, the Space Schedule 601 and the Input Schedule 602. The Space Schedule tracks all currently recorded programs and those which have been scheduled to be recorded in the future. The amount of space available at any given moment in time may be found by generating the sum of all occupied space (or space that will be occupied at that time) and subtracting that from the total capacity available to store programs. Programs scheduled for recording based on inferred preferences ("fuzzy" recordings) are not counted in this calculation; such programs automatically lose all conflict decisions.

A program may be recorded 603 if at all times between when the recording would be initiated and when it expires, sufficient space is available to hold it. In addition, for the duration of the program, there must be an input available from

which to record it. The Input Schedule 602 tracks the free and occupied time slots for each input source. In a preferred embodiment of the invention, the input sources may not be used for identical services, *e.g.*, one input may be from a digital television signal and another from an analog television signal with different programming. In this case, only those inputs from which the desired program can be recorded are considered during scheduling.

With respect to Fig 7, a flowchart is shown describing the steps taken to schedule a recording in the preferred embodiment. First, an ordered list of showings of the program of interest are generated 701. Although a preferred embodiment of the invention orders these showings by time, such that the recording is made as soon as possible, any particular ordering might be chosen. Each showing in this list 702 is then checked to see if input 703 or space 704 conflicts occur as described above. If a showing is found with no conflicts, then the program is scheduled for recording 705.

Otherwise, a preferred embodiment of the invention selects only those showings of the program which have no input conflicts 706. Referring again to Fig. 6, one can see that over the lifetime of a recording the amount of available space will vary as other programs are recorded or expire. The list of showings is then sorted, preferably by the minimum amount of available space during the lifetime of the candidate recording. Other orderings may be chosen.

Referring again to Fig. 7, for each candidate showing, the viewer is presented with the option of shortening the expiration dates on conflicting programs 708, 709. This ordering results in the viewer being presented these choices in order from least impact on scheduled programs to greatest 707; there is no requirement of the invention that this ordering be used versus any other.

Should the viewer reject all opportunities to shorten expiration times, the final step involves selecting those showings with input conflicts 710, and sorting these showings as in the first conflict resolution phase 711. The viewer is then presented with the option to cancel each previously scheduled recording in favor of the desired program 712, 713. Of course, the viewer may ultimately decide that nothing new will be recorded 714.

In a preferred embodiment of the invention, all conflicts are resolved as early as possible, giving the viewer more control over what is recorded. When the viewer makes an explicit selection of a program to record, the algorithm described in Fig. 7 is used to immediately schedule the recording and manage any conflicts that arise.

Once an explicit selection has been made, and the viewer informed that the recording will be done, it will not be canceled without explicit approval of the viewer.

Fuzzy recordings are periodically scheduled by a background task on the client device. Given the prioritized list of preferred programs as described earlier, the background scheduler attempts to schedule each preferred program in turn until the list is exhausted or no further opportunity to record is available. A preferred program is scheduled if and only if there are no conflicts with other scheduled programs. A preferred program which has been scheduled may be deleted under two conditions: first, if it conflicts with an explicit selection, and second, if a change in viewer preferences identifies a higher priority program that could be recorded at that time.

A further complication arises when handling aggregate viewing objects for which recording is requested. If conflict resolution was handled according to the method above for such objects, a potentially large number of conflicts might be generated, leading to a confusing and frustrating experience for the viewer in resolving the conflicts. Thus, when aggregate objects are chosen for recording, conflicts are automatically resolved in favor of the existing schedule.

In a preferred embodiment of the invention, conflicts resulting from the recording of aggregate objects will be resolved using the preference weighting of the programs involved; if multiple conflicts are caused by a particular program in the aggregate object, it will only be recorded if its preference exceeds that of all conflicting programs.

Methods Applied to Software Objects

The client system requires a complex software environment for proper operation. An operating system manages the interaction between hardware devices in the client and software applications which manipulate those devices. The television viewing object database is managed by a distinct software application. The time-warping software application is yet another application.

It is desirable to add new features or correct defects in these and other software subsystems which run on the client hardware device. Using the methods described herein, it is possible to replicate viewing objects containing updated software modules into the client system database. Once present in the client system database, the following unique and novel methods are used to install the updated software and cause the client system to begin executing the new software.

The software environment of the device is instantiated as a sequence of steps that occur when power is first applied to the device, each step building up state information which supports proper application of the following step. The last step launches the applications which manage the device and interact with the viewer. These steps are:

1. A read-only or electrically programmable memory in the device holds an initial bootstrap sequence of instructions. These instructions initialize low-level parameters of the client device, initialize the disk storage system, and load a bootstrap loader from the disk into memory, to which execution is then passed. This initial bootstrap may be changed if it resides in an electrically programmable memory.
2. The second stage boot loader then locates the operating system on the disk drive, loads the operating system into memory, and passes execution to the operating system. This loader must exist at a specific location on the disk so as to be easily located by the initial loader.

The operating system performs necessary hardware and software initialization. It then loads the viewing object database software from the disk drive, and begins execution of the application. Other application software, such as the time-warping software and viewer interaction software, are also loaded and started. This

software is usually located in a separate area on the disk from the object database or captured television programs.

Ideally, new software would be installed by simply copying it to the appropriate place on the disk drive and rebooting the device. This operation is fraught with danger, especially in a home environment. Power may fail while copying the software, resulting in an inconsistent software image and potential operating problems. The new software may have defects which prevent proper operation. A failure may occur on the disk drive, corrupting the software image.

Although the methods of this invention have referred to a disk drive, one skilled in the art will readily appreciate that the methods described here apply generally to any persistent storage system. A disk drive, and other persistent storage systems, are typically formatted into a sequence of fixed-size blocks, called sectors. "Partitions" are sequential, non-overlapping subsets of this sequence which break up the storage into logically independent areas.

With respect to Fig. 8, the invention maintains a sector of information at a fixed location on the disk drive 803 called the "boot sector" 804. The boot sector 804 contains sufficient information for the initial bootstrap 801 to understand the partitioning of the drive 803, and to locate the second stage boot loader 806.

The disk is partitioned into at least seven (7) partitions. There are two (2) small partitions dedicated to holding a copy of the second stage boot loader 806, two (2) partitions holding a copy of the operating system kernel 807, two (2) partitions containing a copy of the application software 808, and a partition to be used as scratch memory 809. For duplicated partitions, an indication is recorded in the boot sector 805 in which one of the partitions is marked "primary", and the second is marked "backup".

One skilled in the art will readily appreciate that, although two partitions are described herein for redundancy, triple, quadruple or greater degrees of redundancy can be achieved by creating more duplicated partitions.

With respect to Figs. 9a and 9b, on boot 901, the initial bootstrap code reads the boot sector 902, scans the partition table and locates the "primary" partition for the second stage boot loader. It then attempts to load this program into memory 903. If it fails 904, for instance, due to a failure of the disk drive, the boot loader attempts to load the program in the "backup" partition into memory 905. Whichever attempt succeeds, the boot loader then passes control to the newly loaded program, along with an indication of which partition the program was loaded from 906.

Similarly, the second stage boot loader reads the partition table and locates the "primary" operating system kernel 907. If the kernel can not be loaded 908, the "backup" kernel is loaded instead 909. In any case, control is passed to the operating system along with an indication of the source partition, along with the passed source partition from above 910.

Finally, the operating system locates the "primary" partition containing application software and attempts to load the initial application 911. If this fails 912, then the operating system locates the "backup" partition and loads the initial application from it 913. An indication of the source partition is passed to the initial application, along with the source partition information from the previous steps. At this point, application software takes over the client system and normal viewing management behavior begins 914.

This sequence of operations provides a reasonable level of protection from disk access errors. It also allows for a method which enables new software at any of these levels to be installed and reliably brought into operation.

An "installer" viewing object in the object database is used to record the status of software installation attempts. It records the state of the partitions for each of the three levels above, including an indication that an attempt to install new software is underway 915. This operation is reliable due to the transactional nature of the database.

Referring to Fig. 10, installing a new software image at any of the three levels is handled as follows: the new software image is first copied into the appropriate

backup partition 1001, and an indication is made in the database that a software installation is underway 1002. The primary and backup partition indications in the partition table are then swapped 1003, and the system rebooted 1004. Eventually, control will be passed to the initial application.

Referring again to Fig. 9b, the first task of this application is to update the installer object. For each level 921, 922, the application checks if an installation was in process 916, 917, and verifies that the level was loaded off of the primary partition 918. If so, the installation at that level was successful, and the installer object is updated to indicate success for that level 919. Otherwise, the application copies the backup partition for that level over the primary partition and indicates failure in the installer object for that level 920. Copying the partition insures that a backup copy of known good software for a level is kept available at all times.

In a preferred embodiment of the invention, finalization of the installation for the top application level of software may be delayed until all parts of the application environment have been successfully loaded and started. This provides an additional level of assurance that all parts of the application environment are working properly before permanently switching to the new software.

Methods Applied to Operations Status Objects

Operations status objects are a class of viewing object in which information about the usage, performance and behavior of the client system is recorded. These objects are collected by the central site whenever communication with the central site is established.

The following operations status indicators are recorded for later collection along with a time stamp:

1. Viewer actions, primarily pressing buttons on a remote control device, are recorded. Each "button press" is recorded along with the current time, and any other contextual information, such as the current viewer context. Post-

processing of this object at the central site results in a complete trace of viewer actions, including the context in which each action is taken.

2. Automatic actions, such as beginning or ending the recording of a program, or choosing a program to record based on viewer preferences, are recorded. In addition, deletion of captured programs is recorded. Post-processing of this object at the central site results in a complete trace of program capture actions taken by the client system, including the programs residing in the persistent store at any point in time.

3. Software installation actions, including reception, installation, and post-reboot results are recorded.

4. Hardware exceptions of various kinds, including but not limited to: power fail/restart, internal temperature profile of the device, persistent storage access errors, memory parity errors and primary partition failures.

Since all actions are recorded along with a time stamp, it is possible to reconstruct the behavior of the client system using a linear time-based ordering. This allows manual or automatic methods to operate on the ordered list of events to correlate actions and behaviors. For instance, if an expected automatic action does not occur soon after rebooting with new software, it may be inferred that the new software was defective.

Processing of Television Viewing Objects by Central Site Systems

Sources of Television Viewing Objects

A client system has a single source of television viewing objects: the central site. The central site object database has many sources of television viewing objects:

1. Program guide information obtained from outside sources is processed to produce a consistent set of program guide objects, indicating "programs",

"showings", "channels", "networks" and other related objects. This set of objects will have dependencies ("channels" depend on "networks", "showings" depend on "programs") and other interrelationships. When a complete, consistent set of objects is ready, it is added to the database as an atomic operation.

2. New software, including new applications or revisions of existing software, are first packaged into "software" viewing objects. As above, the software may have interdependencies, such as an application depending on a dynamically loaded library, which must be reflected in the interrelationships of the software objects involved. In another example, there may be two types of client systems in use, each of which requires different software objects; these software objects must have attributes present indicating the type of system they are targeted at. Once a consistent set of objects is available, it is added to the database as an atomic operation.

3. Each client system has a unique, secret key embedded within it. The public key matching this secret key is loaded into a "client" management object, along with other interesting information about the client, such as client type, amount of storage in the system, etc. These objects are used to generate authentication objects as necessary.

4. Aggregation program guide objects are added in a similar fashion. In this case, however, the aggregation object must refer to primitive program guide objects already present in the database. Also attached to the aggregation object are other objects, such as a textual description, a screen-based icon, and other informational attributes. Once a consistent set of ancillary objects to the aggregation is available, it is added to the database as an atomic operation.

5. Data collected from client systems.

It should be clear that there may be any number of sources of viewing objects, and this enumeration simply shows the most basic possible sources.

Operations on Television Viewing Objects

There are a large number of possible operations on the central television viewing object database. The following examples are meant to show the type of processing that may be performed, however the potential operations are not limited to these examples:

1. Using various viewing objects, a number of interesting statistical analysis tasks may be performed:
   1.1. By examining large numbers of uploaded operations status objects, it is possible to perform extensive analysis of hardware reliability trends and failure modes. For instance, it is possible to correlate internal temperature with expected MTBF (Mean Time Between Failures) of client devices.
   1.2. By examining large numbers of uploaded viewing information, it is possible to derive demographic or psychographic information about various populations of client devices. For example, it is possible to correlate TV programs most watched within specific zip codes in which the client devices reside.
   1.3. Similarly, by examining large numbers of viewing information objects, it is possible to generate "rating" and "share" values for particular programs with fully automated methods, unlike existing program rating methods.
   1.4. There are many other examples of statistical analysis tasks that might be performed on the viewing object database; these examples are not meant to limit the applicability of the invention, but to illustrate by example the spectrum of operations that might be performed.
2. Specialty aggregation objects may be automatically generated based on one or more attributes of all available viewing objects.

   Such generation is typically performed by first extracting information of interest from each viewing object, such as program description, actor, director, etc., and constructing a simple table of programs and attributes. An aggregate viewing object is then generated by choosing one or more attributes, and adding to the aggregate those programs for which the chosen attributes match in some way.

   These objects are then included in the slices generated for transmission,

possibly based on geographic or other information. Some example aggregates that might be created are:

2.1. Aggregates based on events, such as a major league football game in a large city. In this case, all programs viewable by client devices in or around that city are collected, and the program description searched for the names of the teams playing, coaches names, major player's names, the name of the ballpark, etc. Matching program objects are added to the aggregate, which is then sliced for transmission only to client devices in regions in and around the city.

2.2. Aggregates based on persons of common interest to a large number of viewers. For instance, an aggregate might be constructed of all "John Wayne" movies to be broadcast in the next week.

2.3. Aggregates based on viewing behavior can be produced. In this case, uploaded viewing objects are scanned for elements of common interest, such as types of programs viewed, actual programs viewed, etc. For example, a "top ten list" aggregate of programs viewed on all client devices in the last week might be generated containing the following week's showing of those programs.

2.4. Aggregates based on explicit selections by viewers. During viewing of a program, the viewer might be presented with an opportunity to "vote" on the current program, perhaps on the basis of four perceived attributes (storyline, acting, directing, cinematography), which generates viewing objects that are uploaded later. These votes are then scanned to determine an overall rating of the program, which is transmitted to those who voted for their perusal.

2.5. There are many other examples of how the basic facilities of this invention allow the service operator to provide pre-sorted and pre-selected groups of related programs to the user of the client device for perusal and selection. These examples are not meant to limit the applicability of the invention, but to illustrate by example the spectrum of operations that might be performed.

3. Manual methods may also be used to generate aggregate objects, a process sometimes called "authoring". In this case, the person creating the

aggregate chooses programs for explicit addition to the aggregate. It is then transmitted in the same manner as above.

Clearly, aggregation program objects may also permit the expression of preferences or recording of other information. These results may be uploaded to the central site to form a basis for the next round of aggregate generation or statistical analysis, and so on.

This feedback loop closes the circuit between service provider and the universe of viewers using the client device. This unique and novel approach provides a new form of television viewing by providing unique and compelling ways for the service provider to present and promote the viewing of television programs of interest to individuals while maintaining reliable and consistent operation of the service.

Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention. Accordingly, the invention should only be limited by the Claims included below.

CLAIMS

1. A process for scheduling the recording, storing, and deleting of television and Web page program material on a storage medium in a computer environment, comprising the steps of:

accepting as input a prioritized list of program viewing preferences;

comparing said list with the database of program guide objects;

generating a schedule of time versus available storage space that is optimal for the viewer's explicit or derived preferred programs;

wherein said preferred programs include television broadcast programs and Universal Resource Locators (URLs); and

wherein said program guide objects indicate when programs of interest are actually broadcast.

2. The process of claim 1, wherein the viewer may request that certain programs be captured, which results in the highest possible priority for those programs.

3. The process of claim 1, wherein the viewer may explicitly express preferences using appurtenances provided through the viewer interface.

4. The process of claim 1, wherein said preferences may be inferred from viewing patterns.

5. The process of claim 1, wherein said preferences correspond to television viewing objects stored in a replicated database.

6. The process of claim 1, further comprising the step of:

providing a space schedule;

providing an input schedule;

wherein said space schedule tracks all currently recorded programs and the programs that have been scheduled to be recorded in the future; and

wherein said input schedule tracks the free and occupied time slots for each input source.

7. The process of claim 6, wherein the amount of space available at any given moment in time may be found by generating the sum of all occupied

space or space that will be occupied at that particular time, and subtracting that from the total capacity available to store programs.

8. The process of claim 1, wherein programs scheduled for recording based on inferred preferences automatically lose all conflict decisions.

9. The process of claim 1, wherein a program is recorded if at all times between when the recording would be initiated and when it expires, sufficient space is available to hold it.

10. The process of claim 6, wherein there must be an input available from which to record for the duration of the program.

11. The process of claim 6, wherein only those inputs from which the desired program can be recorded are considered during scheduling.

12. The process of claim 1, further comprising the step of:
generating an ordered list of showings of the program of interest.

13. The process of claim 12, wherein each showing in said list is checked to see if input or space conflicts occur.

14. The process of claim 12, wherein if a showing is found with no conflicts, then the program is scheduled for recording for said showing.

15. The process of claim 12, further comprising the step of:
sorting said list of showings; and
wherein the ordering of said list results in the viewer being presented with any conflicting programs in order from least impact on scheduled programs to greatest.

16. The process of claim 15, wherein for each candidate showing in said list, the viewer is presented with the option of shortening the expiration dates on conflicting programs.

17. The process of claim 15, wherein the viewer is presented with the option to cancel each previously scheduled recording that has an input conflict with the desired program.

18. The process of claim 1, further comprising the step of:
providing a background scheduler.

19. The process of claim 18, wherein said background scheduler schedules each preferred program in turn until the list of preferred programs is exhausted or no further opportunity to record is available.

20. The process of claim 18, wherein a preferred program is scheduled if and only if there are no conflicts with other scheduled programs.

21. The process of claim 18, wherein a preferred program which has been scheduled may be deleted if it conflicts with an explicit selection or if a change in viewer preferences identifies a higher priority program that could be recorded at that time.

22. The process of claim 1, wherein all conflicts are resolved as early as possible.

23. The process of claim 1, wherein any schedule conflicts are determined immediately when the viewer makes an explicit selection of a program to record.

24. The process of claim 1, wherein if there are schedule conflicts with other programs that the viewer has explicitly selected, the viewer is asked which recordings should be canceled and which should be completed.

25. The process of claim 4, wherein schedule conflicts between explicit program selections and inferred "fuzzy" program selections are resolved in favor of said explicit selections without asking the viewer.

26. The process of claim 1, wherein the expiration time of any conflicting stored programs is shortened to exactly that needed to allow recording of a desired program.

27. The process of claim 1, wherein schedule conflicts resulting from the recording of aggregate objects are resolved using the preference weighting of the programs involved.

28. The process of claim 1, wherein if multiple conflicts are caused by a particular program in an aggregate object, it will only be recorded if its preference exceeds that of all conflicting programs.

29. An apparatus for scheduling the recording, storing, and deleting of television and Web page program material on a storage medium in a computer environment, comprising:

a module for accepting as input a prioritized list of program viewing preferences;

a module for comparing said list with the database of program guide objects;

a module for generating a schedule of time versus available storage space that is optimal for the viewer's explicit or derived preferred programs;

wherein said preferred programs include television broadcast programs and Universal Resource Locators (URLs); and

wherein said program guide objects indicate when programs of interest are actually broadcast.

30. The apparatus of claim 29, wherein the viewer may request that certain programs be captured, which results in the highest possible priority for those programs.

31. The apparatus of claim 29, wherein the viewer may explicitly express preferences using appurtenances provided through the viewer interface.

32. The apparatus of claim 29, wherein said preferences may be inferred from viewing patterns.

33. The apparatus of claim 29, wherein said preferences correspond to television viewing objects stored in a replicated database.

34. The apparatus of claim 29, further comprising:

a space schedule;

an input schedule;

wherein said space schedule tracks all currently recorded programs and the programs that have been scheduled to be recorded in the future; and

wherein said input schedule tracks the free and occupied time slots for each input source.

35. The apparatus of claim 34, wherein the amount of space available at any given moment in time may be found by generating the sum of all occupied space or space that will be occupied at that particular time, and subtracting that from the total capacity available to store programs.

36. The apparatus of claim 29, wherein programs scheduled for recording based on inferred preferences automatically lose all conflict decisions.

37. The apparatus of claim 29, wherein a program is recorded if at all times between when the recording would be initiated and when it expires, sufficient space is available to hold it.

38. The apparatus of claim 34, wherein there must be an input available from which to record for the duration of the program.

39. The apparatus of claim 34, wherein only those inputs from which the desired program can be recorded are considered during scheduling.

40. The apparatus of claim 29, further comprising:

a module for generating an ordered list of showings of the program of interest.

41. The apparatus of claim 40, wherein each showing in said list is checked to see if input or space conflicts occur.

42. The apparatus of claim 40, wherein if a showing is found with no conflicts, then the program is scheduled for recording for said showing.

43. The apparatus of claim 40, further comprising:

a module for sorting said list of showings; and

wherein the ordering of said list results in the viewer being presented with any conflicting programs in order from least impact on scheduled programs to greatest.

44. The apparatus of claim 43, wherein for each candidate showing in said list, the viewer is presented with the option of shortening the expiration dates on conflicting programs.

45. The apparatus of claim 43, wherein the viewer is presented with the option to cancel each previously scheduled recording that has an input conflict with the desired program.

46. The apparatus of claim 29, further comprising:
a background scheduler.

47. The apparatus of claim 46, wherein said background scheduler schedules each preferred program in turn until the list of preferred programs is exhausted or no further opportunity to record is available.

48. The apparatus of claim 46, wherein a preferred program is scheduled if and only if there are no conflicts with other scheduled programs.

49. The apparatus of claim 46, wherein a preferred program which has been scheduled may be deleted if it conflicts with an explicit selection or if a change in viewer preferences identifies a higher priority program that could be recorded at that time.

50. The apparatus of claim 29, wherein all conflicts are resolved as early as possible.

51. The apparatus of claim 29, wherein any schedule conflicts are determined immediately when the viewer makes an explicit selection of a program to record.

52. The apparatus of claim 29, wherein if there are schedule conflicts with other programs that the viewer has explicitly selected, the viewer is asked which recordings should be canceled and which should be completed.

53. The apparatus of claim 32, wherein schedule conflicts between explicit program selections and inferred "fuzzy" program selections are resolved in favor of said explicit selections without asking the viewer.

54. The apparatus of claim 29, wherein the expiration time of any conflicting stored programs is shortened to exactly that needed to allow recording of a desired program.

55. The apparatus of claim 29, wherein schedule conflicts resulting from the recording of aggregate objects are resolved using the preference weighting of the programs involved.

56. The apparatus of claim 29, wherein if multiple conflicts are caused by a particular program in an aggregate object, it will only be recorded if its preference exceeds that of all conflicting programs.

57. A program storage medium readable by a computer, tangibly embodying a program of instructions executable by the computer to perform method steps for scheduling the recording, storing, and deleting of television and Web page program material on a storage medium in a computer environment, comprising the steps of:

accepting as input a prioritized list of program viewing preferences;

comparing said list with the database of program guide objects;

generating a schedule of time versus available storage space that is optimal for the viewer's explicit or derived preferred programs;

wherein said preferred programs include television broadcast programs and Universal Resource Locators (URLs); and

wherein said program guide objects indicate when programs of interest are actually broadcast.

58. The method of claim 57, wherein the viewer may request that certain programs be captured, which results in the highest possible priority for those programs.

59. The method of claim 57, wherein the viewer may explicitly express preferences using appurtenances provided through the viewer interface.

60. The method of claim 57, wherein said preferences may be inferred from viewing patterns.

61. The method of claim 57, wherein said preferences correspond to television viewing objects stored in a replicated database.

62. The method of claim 57, further comprising the step of:

providing a space schedule;

providing an input schedule;

wherein said space schedule tracks all currently recorded programs and the programs that have been scheduled to be recorded in the future; and

wherein said input schedule tracks the free and occupied time slots for each input source.

63. The method of claim 62, wherein the amount of space available at any given moment in time may be found by generating the sum of all occupied space or space that will be occupied at that particular time, and subtracting that from the total capacity available to store programs.

64. The method of claim 57, wherein programs scheduled for recording based on inferred preferences automatically lose all conflict decisions.

65. The method of claim 57, wherein a program is recorded if at all times between when the recording would be initiated and when it expires, sufficient space is available to hold it.

66. The method of claim 62, wherein there must be an input available from which to record for the duration of the program.

67. The method of claim 62, wherein only those inputs from which the desired program can be recorded are considered during scheduling.

68. The method of claim 57, further comprising the step of:

generating an ordered list of showings of the program of interest.

69. The method of claim 68, wherein each showing in said list is checked to see if input or space conflicts occur.

70. The method of claim 68, wherein if a showing is found with no conflicts, then the program is scheduled for recording for said showing.

71. The method of claim 68, further comprising the step of:

sorting said list of showings; and

wherein the ordering of said list results in the viewer being presented with any conflicting programs in order from least impact on scheduled programs to greatest.

72. The method of claim 71, wherein for each candidate showing in said list, the viewer is presented with the option of shortening the expiration dates on conflicting programs.

73. The method of claim 71, wherein the viewer is presented with the option to cancel each previously scheduled recording that has an input conflict with the desired program.

74. The method of claim 57, further comprising the step of:
providing a background scheduler.

75. The method of claim 74, wherein said background scheduler schedules each preferred program in turn until the list of preferred programs is exhausted or no further opportunity to record is available.

76. The method of claim 74, wherein a preferred program is scheduled if and only if there are no conflicts with other scheduled programs.

77. The method of claim 74, wherein a preferred program which has been scheduled may be deleted if it conflicts with an explicit selection or if a change in viewer preferences identifies a higher priority program that could be recorded at that time.

78. The method of claim 57, wherein all conflicts are resolved as early as possible.

79. The method of claim 57, wherein any schedule conflicts are determined immediately when the viewer makes an explicit selection of a program to record.

80. The method of claim 57, wherein if there are schedule conflicts with other programs that the viewer has explicitly selected, the viewer is asked which recordings should be canceled and which should be completed.

81. The method of claim 60, wherein schedule conflicts between explicit program selections and inferred "fuzzy" program selections are resolved in favor of said explicit selections without asking the viewer.

82. The method of claim 57, wherein the expiration time of any conflicting stored programs is shortened to exactly that needed to allow recording of a desired program.

83. The method of claim 57, wherein schedule conflicts resulting from the recording of aggregate objects are resolved using the preference weighting of the programs involved.

84. The method of claim 57, wherein if multiple conflicts are caused by a particular program in an aggregate object, it will only be recorded if its preference exceeds that of all conflicting programs.

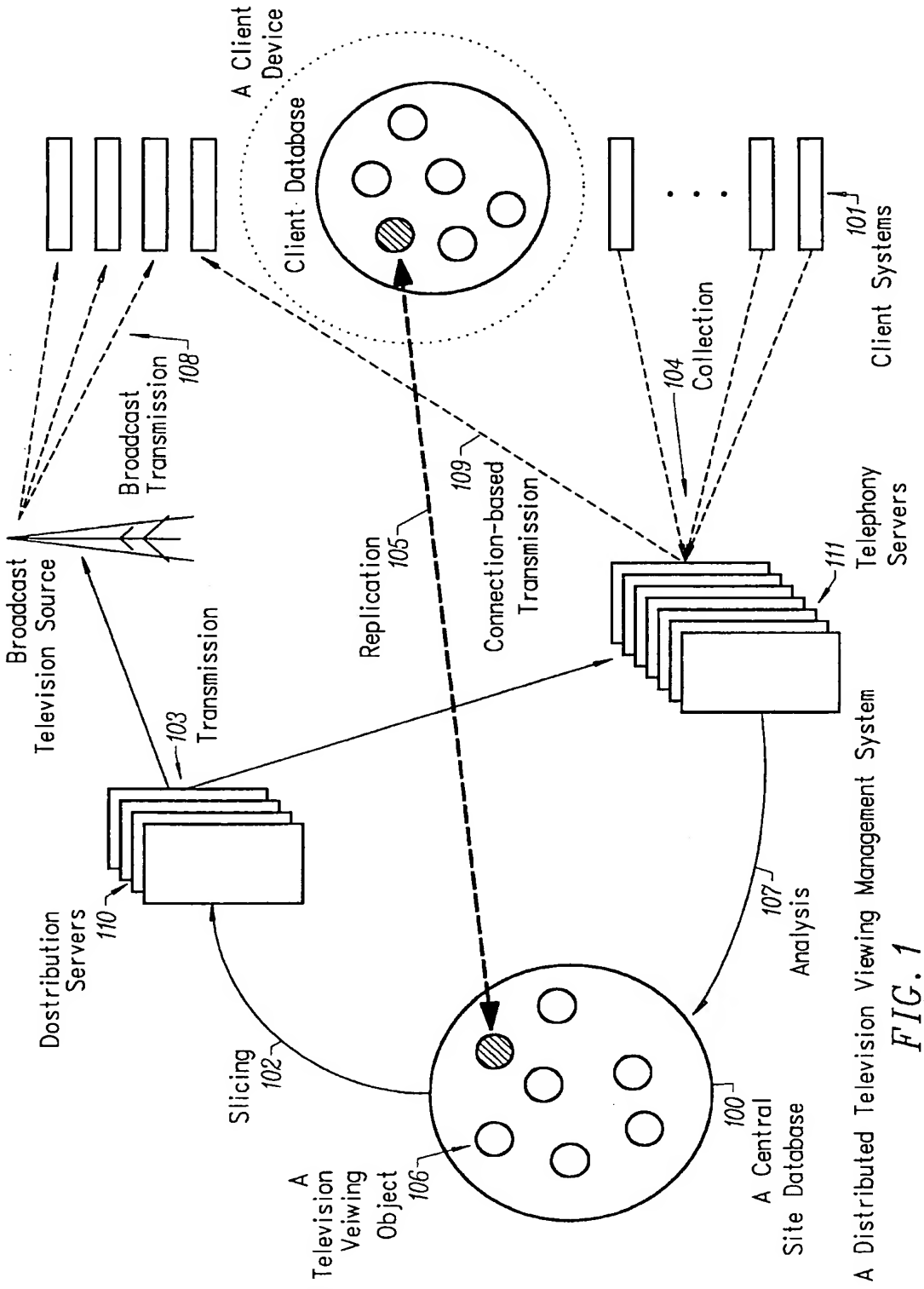
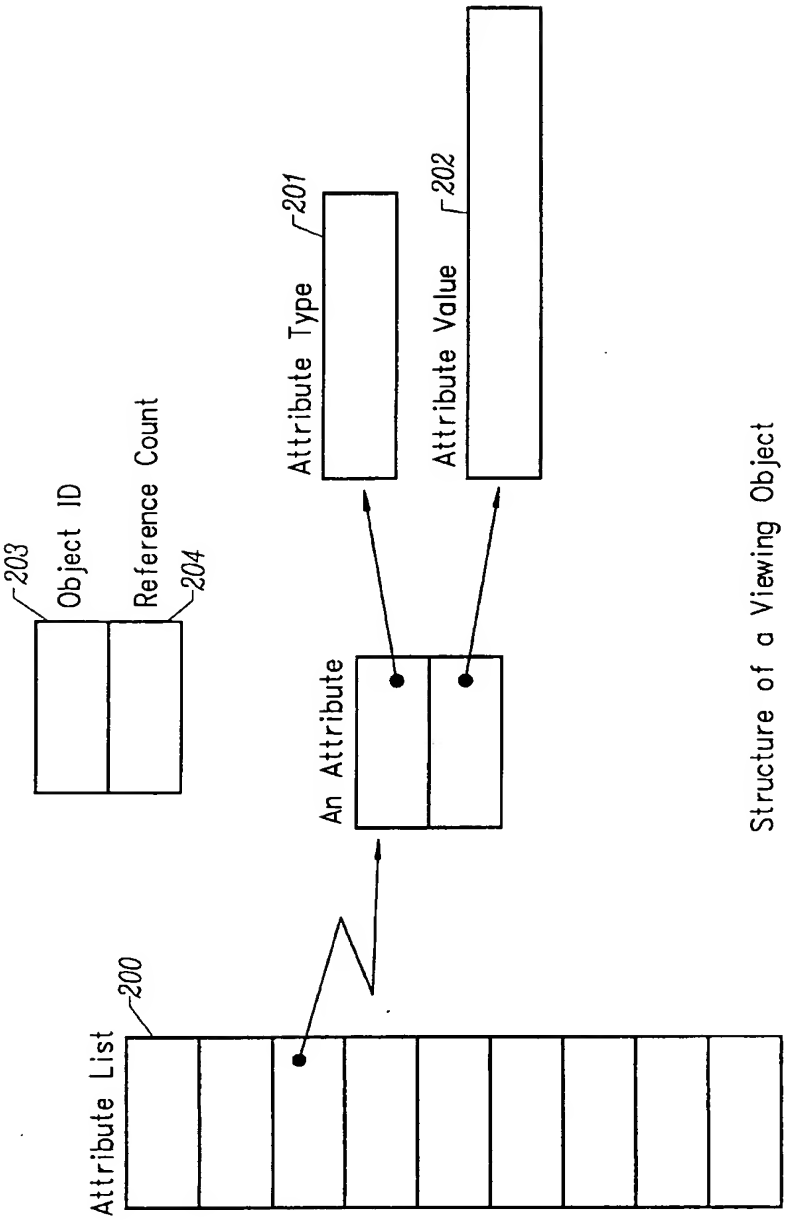
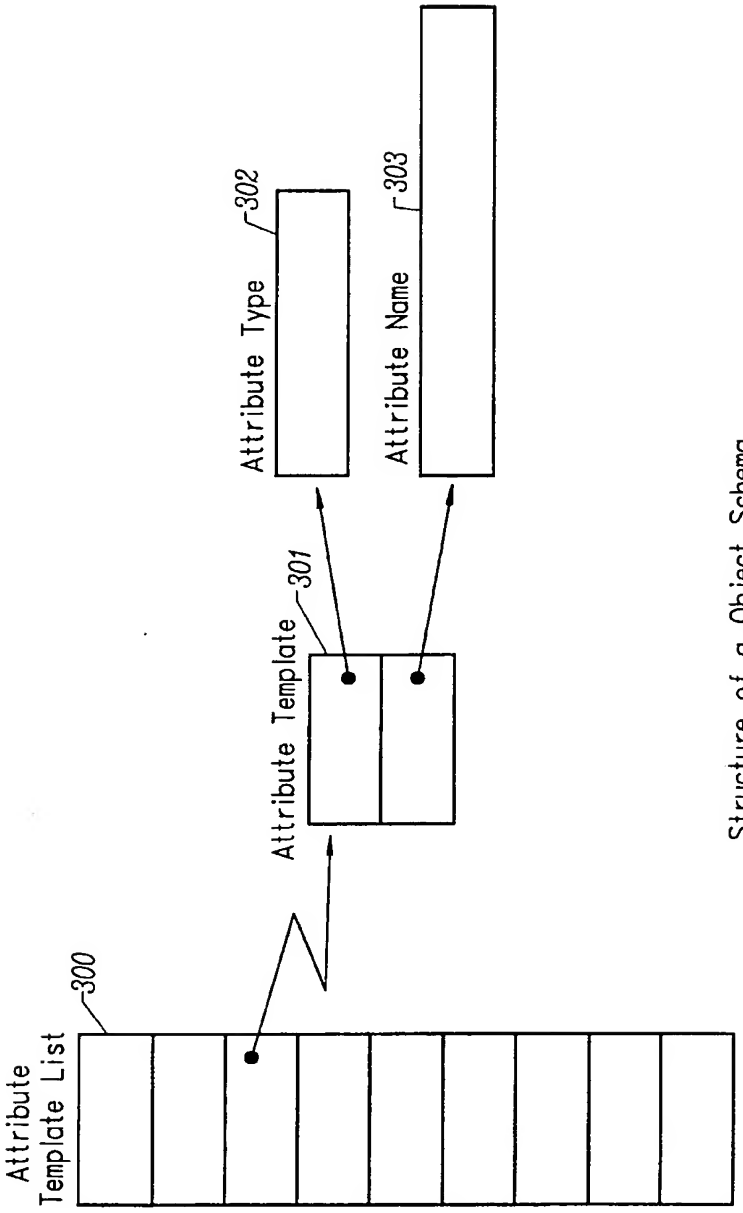

A Client Device
Client Database
Client Systems
101
Broadcast Transmission
108
Broadcast Television Source
Transmission
103
Replication
105
Connection-based Transmission
109
Collection
104
Telephony Servers
111
Dostribution Servers
110
Slicing
102
A Television Veiwing Object
106
A Central Site Database
100
Analysis
107
A Distributed Television Viewing Management System

FIG. 1



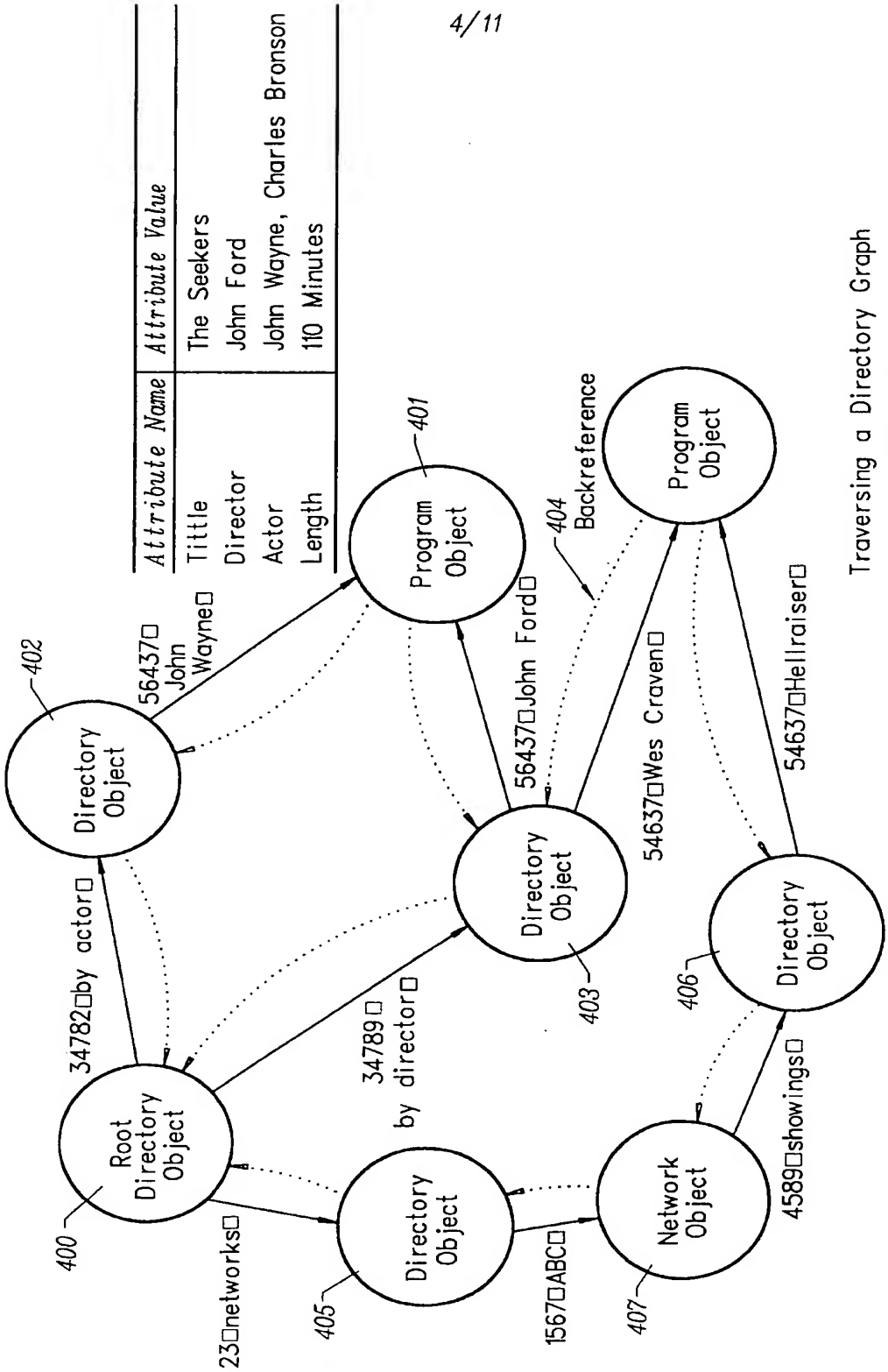
Structure of a Viewing Object

FIG. 2



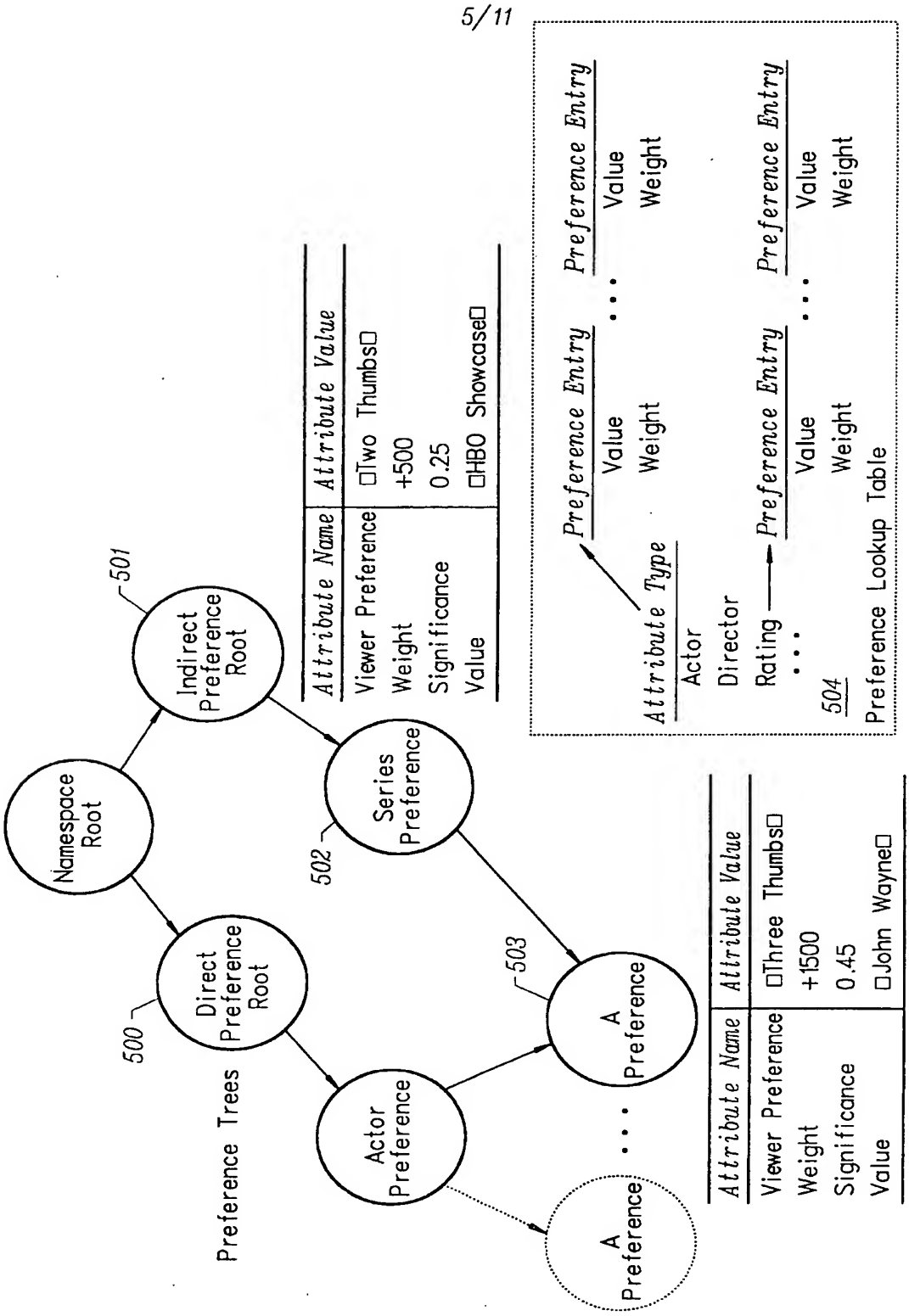
Structure of a Object Schema

FIG. 3



Traversing a Directory Graph

FIG. 4



Processing Preference Information *FIG. 5*

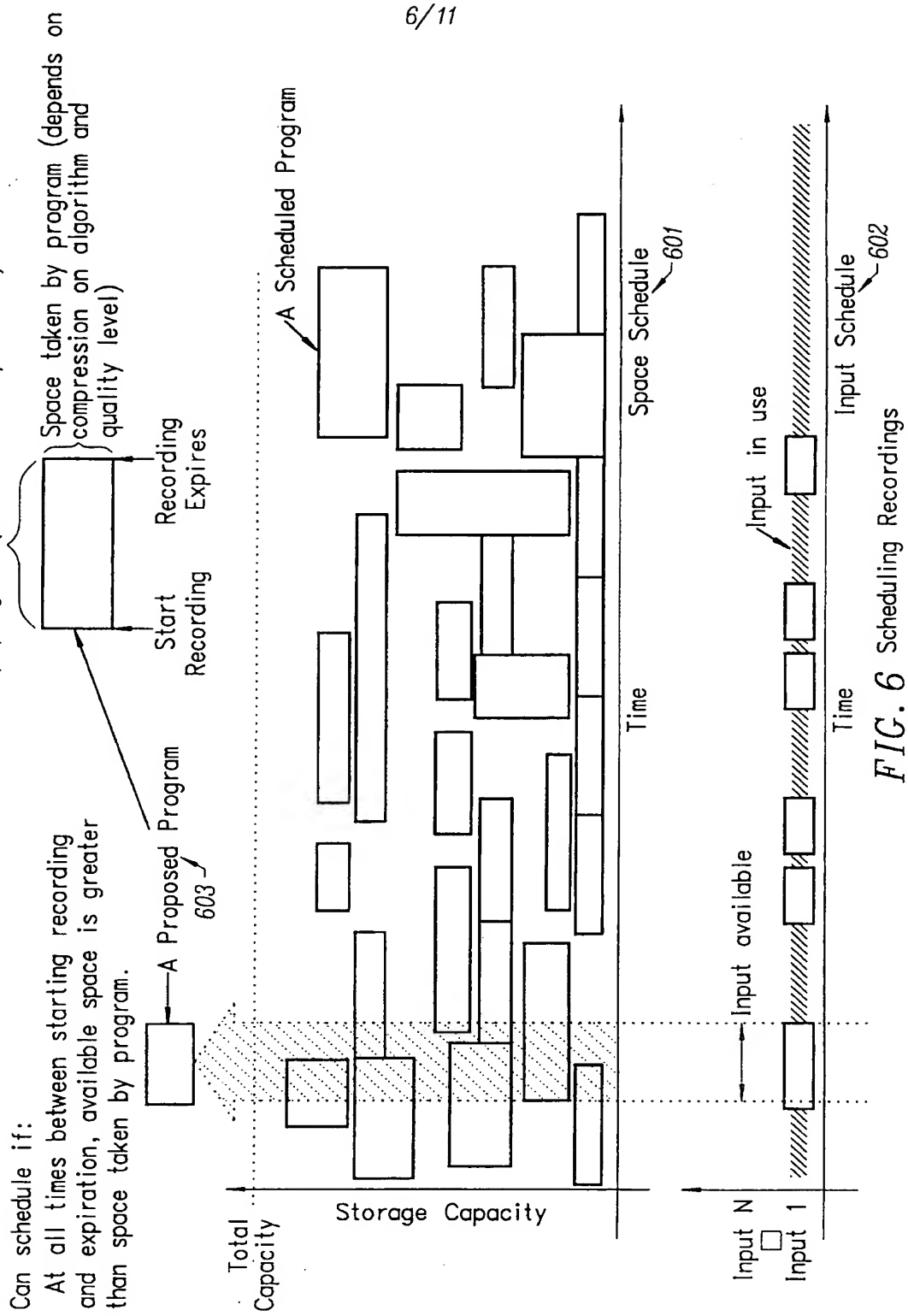
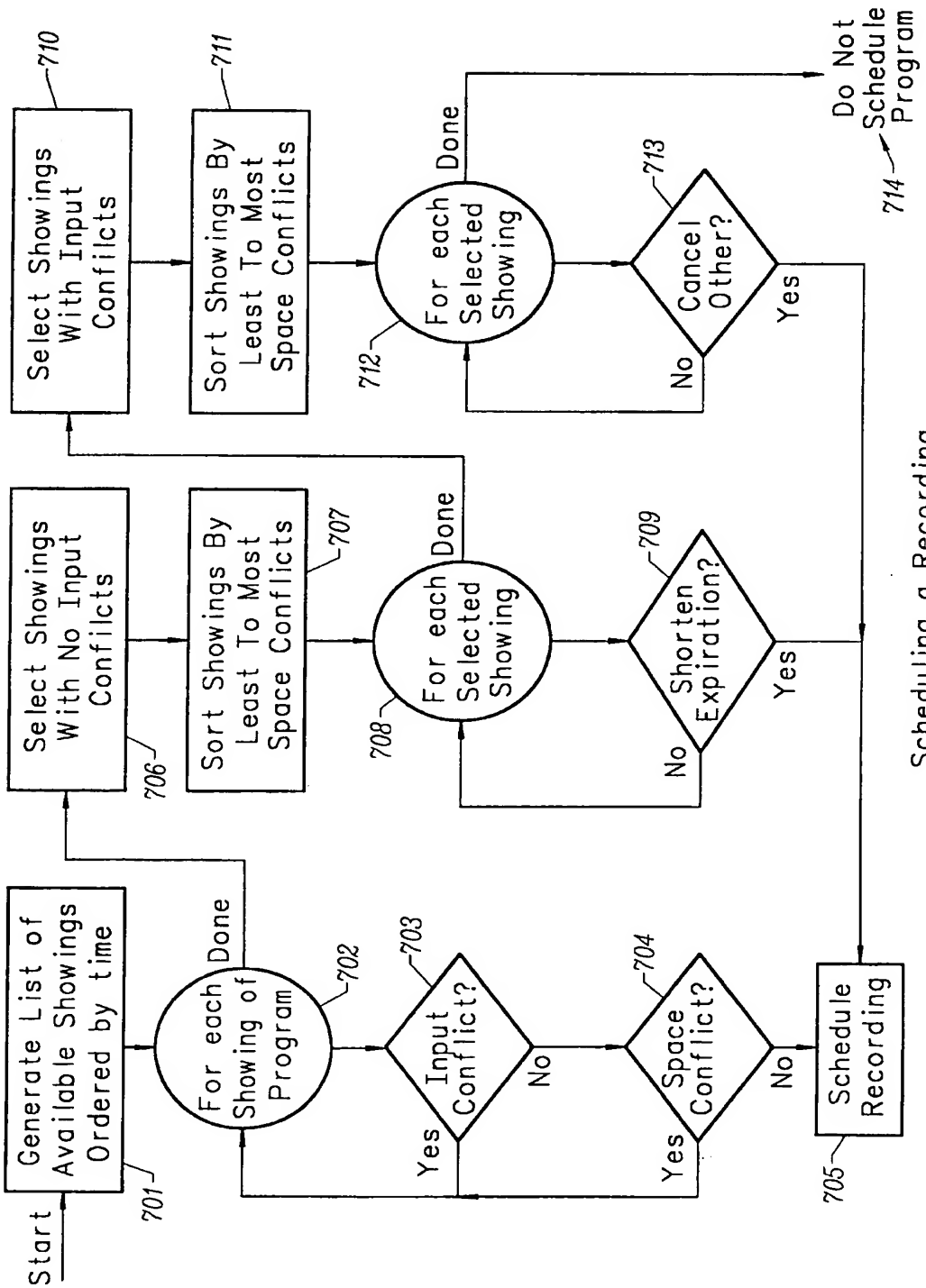


FIG. 6 Scheduling Recordings



Scheduling a Recording

FIG. 7

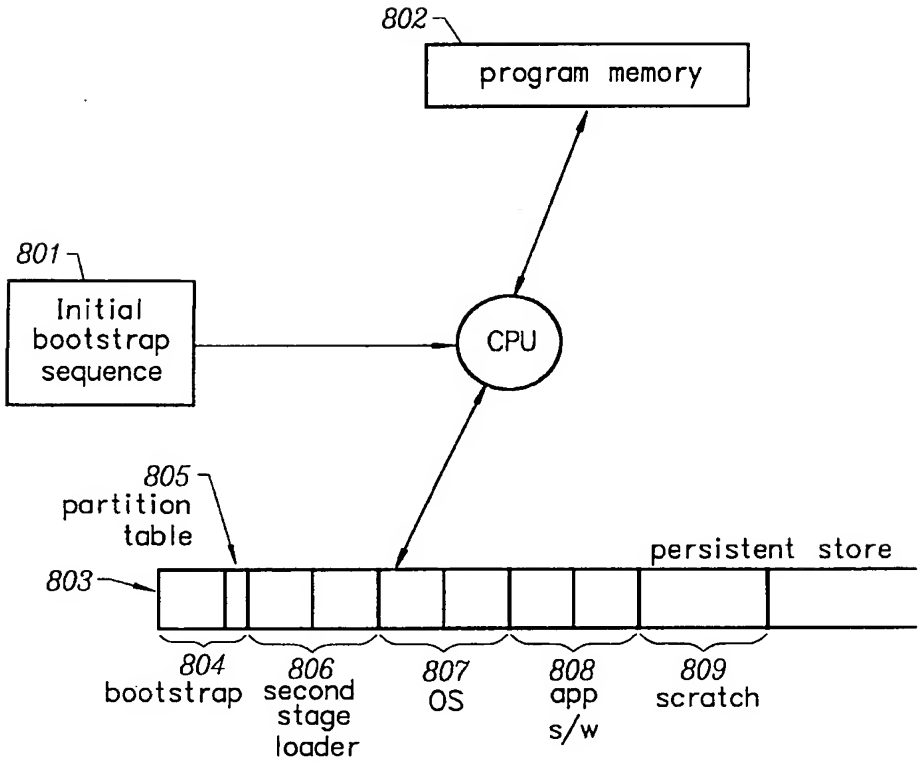

802
program memory
801
Initial bootstrap sequence
CPU
805
partition table
803
persistent store
804
bootstrap
806
second stage loader
807
OS
808
app s/w
809
scratch

FIG. 8

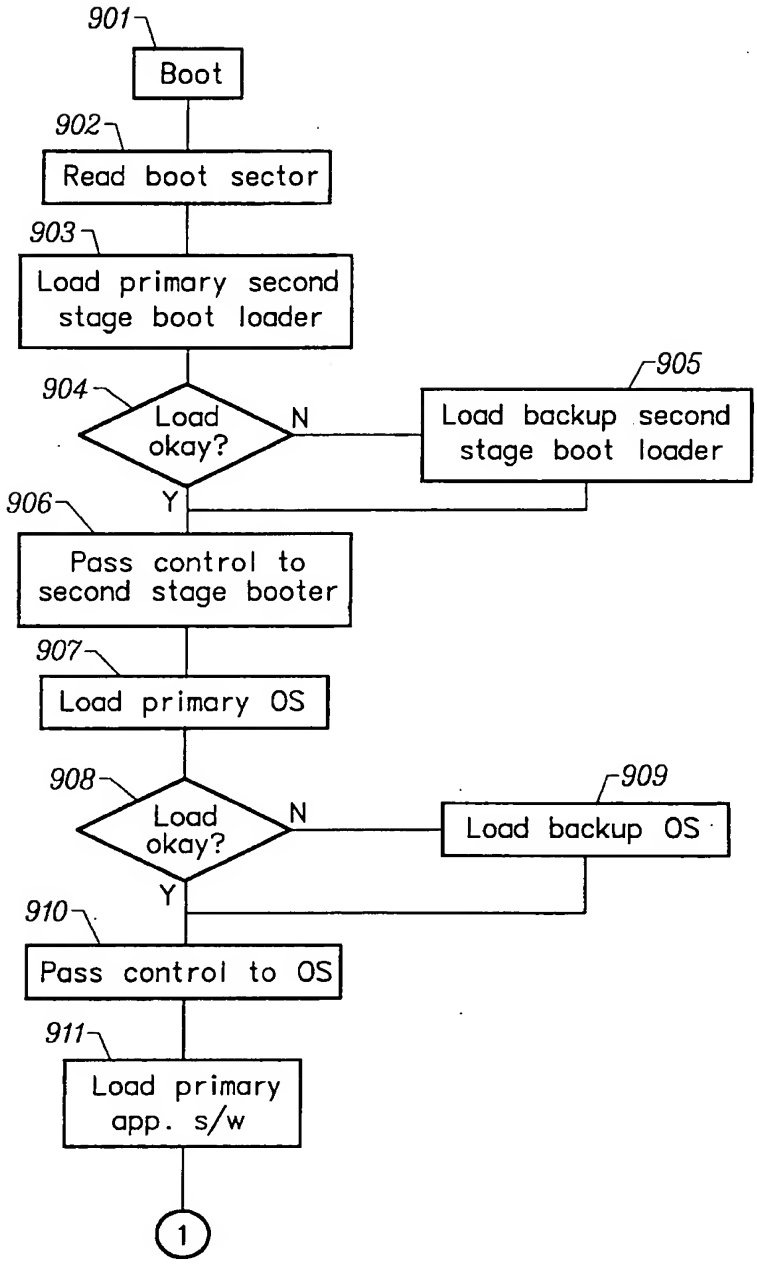

901
Boot
902
Read boot sector
903
Load primary second stage boot loader
904
Load okay?
N
905
Load backup second stage boot loader
Y
906
Pass control to second stage booter
907
Load primary OS
908
Load okay?
N
909
Load backup OS
Y
910
Pass control to OS
911
Load primary app. s/w
1

FIG. 9A

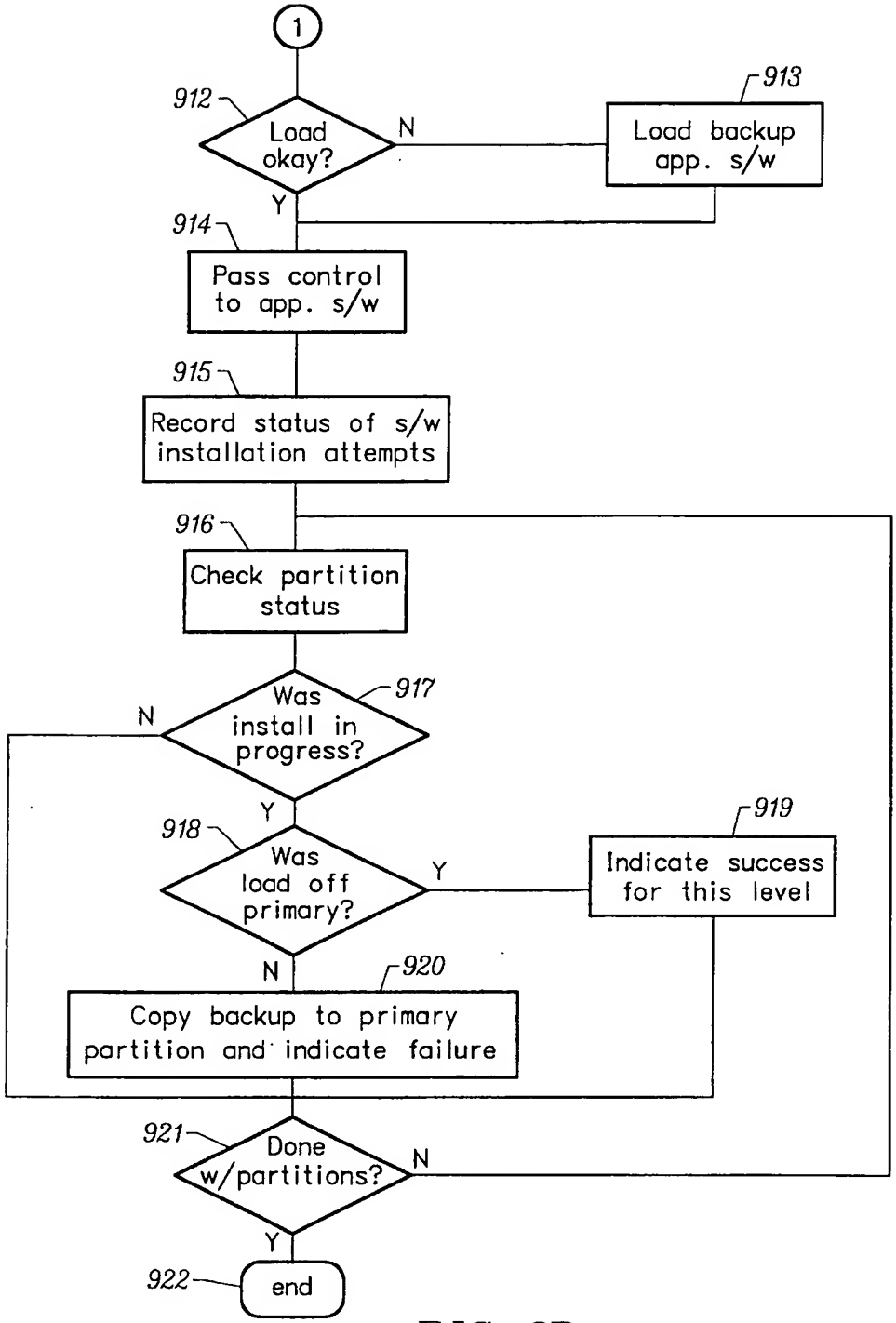

1
912
Load okay?
N
913
Load backup app. s/w
Y
914
Pass control to app. s/w
915
Record status of s/w installation attempts
916
Check partition status
917
Was install in progress?
N
Y
918
Was load off primary?
Y
919
Indicate success for this level
N
920
Copy backup to primary partition and indicate failure
921
Done w/partitions?
N
Y
922
end

FIG. 9B

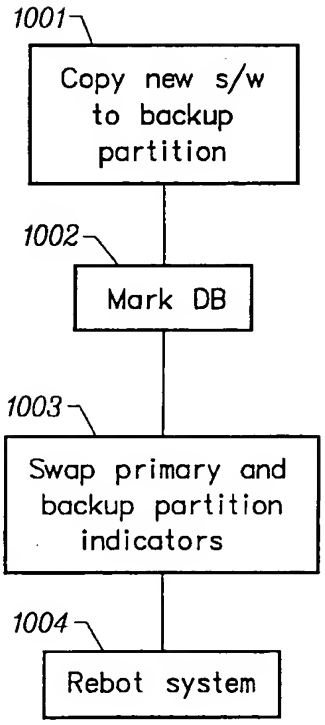

1001
Copy new s/w to backup partition
1002
Mark DB
1003
Swap primary and backup partition indicators
1004
Rebot system

FIG. 10

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/US 00/06473

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N7/16

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
PAJ, WPI Data, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 758 259 A (LAWLER FRANK A) 26 May 1998 (1998-05-26) column 1, line 60 -column 2, line 44	1,29,57
A	WO 98 56188 A (SONY ELECTRONICS INC) 10 December 1998 (1998-12-10) page 4, line 1 -page 5, line 14 page 7, line 1 -page 13, line 25; figures 1-6 page 19, line 4 - line 11 page 23, line 27 -page 25, line 10	1,29,57
A	US 5 758 257 A (HERZ FREDERICK ET AL) 26 May 1998 (1998-05-26) column 4, line 16 -column 6, line 34	1,29,57
A	US 5 774 664 A (HIDARY JACK D ET AL) 30 June 1998 (1998-06-30)	

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 August 2000	18/08/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Fuchs, P

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte ional Application No
PCT/US 00/06473

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 5758259	A	26-05-1998	NONE		
WO 9856188	A	10-12-1998	US	6028600 A	22-02-2000
			US	6072483 A	06-06-2000
			AU	7446798 A	21-12-1998
			EP	0986901 A	22-03-2000
US 5758257	A	26-05-1998	AU	703247 B	25-03-1999
			AU	4410396 A	19-06-1996
			CA	2207868 A	06-06-1996
			EP	0796538 A	24-09-1997
			US	6020883 A	01-02-2000
			WO	9617467 A	06-06-1996
			US	5734720 A	31-03-1998
			US	5754938 A	19-05-1998
			US	5754939 A	19-05-1998
			US	5835087 A	10-11-1998
			US	6088722 A	11-07-2000
			US	6029195 A	22-02-2000
US 5774664	A	30-06-1998	US	5778181 A	07-07-1998
			AU	2070597 A	22-09-1997
			EP	0885525 A	23-12-1998
			EP	0982943 A	01-03-2000
			WO	9733434 A	12-09-1997
			US	6018768 A	25-01-2000

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 935 378 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
11.08.1999 Bulletin 1999/32

(51) Int. Cl.[6]: **H04M 3/42**, H04M 3/50

(21) Application number: **98310516.4**

(22) Date of filing: **21.12.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **16.01.1998 US 8124**

(71) Applicant:
INTERNATIONAL BUSINESS MACHINES CORPORATION
Armonk, NY 10504 (US)

(72) Inventors:
- **Epstein, Mark Edward**
  Katonah, NY 10536 (US)
- **Kanevsky, Dimitri**
  Ossining, NY 10562 (US)
- **Maes, Stephan Herman**
  Danbury, CT06811 (US)

(74) Representative:
Burrington, Alan Graham Headford
Alan Burrington & Associates
41 The Street
Ashtead
Surrey KT21 1AA (GB)

(54) System and methods for automatic call and data transfer processing

(57) A programmable automatic call and data transfer processing system which automatically processes incoming telephone calls, facsimiles and e-mails based on the identity of the caller or author, the subject matter of the message or request, and/or the time of day, which includes: a central server for automatically answering an incoming call and collecting voice data of a caller; a speaker recognition module connected to the server for identifying the caller or author; a switching module responsive to the speaker recognition module for processing the call or message in accordance with a pre-programmed procedure based on the identification of the caller or author; and a programming interface for programming the server, speaker recognizer module and the switching module. The system is programmed by the user to so as to process incoming telephone calls or e-mail and facsimile messages based on the identity of the caller or author, subject matter and content of the message and the time of day. Such processing includes, but is not limited to, switching the call to another system, forwarding the call to another telephone terminal, placing the call on hold, or disconnecting the call. In another aspect of the present invention, the system may be employed to process information retrieved from other telecommunication devices such as voice mail, facsimile/modem or e-mail. The system is capable of tagging the identity of a caller or participants to a teleconference, and transcribing the teleconferences, phone conversations and messages of such callers and participants. The system can automatically index or prioritize the received calls, messages, e-mails and facsimiles according to the caller identification or subject matter of the conversation or message, and allow the user to retrieve messages that either originated from a specific source or caller or retrieve calls which deal with similar or specific subject matter.

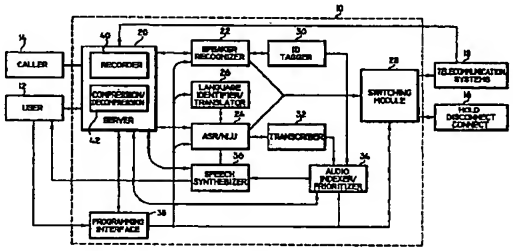


FIG. 2

EP 0 935 378 A2

Description

[0001] The present invention relates to a system and methods for providing automatic call and data transfer processing and, more particularly, to a system and methods for providing automatic call and data transfer processing according to a pre-programmed procedure based on the identity of a caller or author, the subject matter and content of a call or message and/or the time of day of such call or message.

[0002] Generally, in the past, call processing has been manually performed either by a business owner, a secretary or a local central phone service. There are certain conventional devices which partially perform some call processing functions. For example, conventional answering machines and voice-mail services record incoming telephone messages which are then played back by the user of such devices or services. In addition, desktop-telephone software or local PBXs (private branch exchange) provide telephone network switching capabilities. These conventional answering machines, voice-mail services and switching systems, however, are not capable of automatically performing distinct processing procedures that are responsive to the identity of the caller or evaluating the content or subject matter of the call or message and then handling such call or message accordingly. Instead, the user must first answer his or her telephone calls manually, or retrieve such calls from an answering machine or voice-mail, and then decide how to proceed on a call-by-call basis. The present invention eliminates or mitigates such burdensome manual processing.

[0003] Moreover, although protected by Dual Tone Multi-Frequency (DTMF) keying, answering machines and voice-mail services are unable to identify or verify the caller when being remotely accessed or re-programmed by a caller with a valid personal identification number (PIN) which is inputted by DTMF keys. Further, conventional teleconference centers also rely on DTMF PINs for accessibility but are unable to verify and tag the identity of the speaker during a teleconference. Such answering machines, voice-mail and teleconference centers may therefore be breached by unauthorized persons with access to an otherwise valid PIN.

[0004] It is therefore an object of the present invention to provide a system and methods for automatic call and data transfer processing in accordance with a pre-determined manner based on the identity of the caller or author, the subject matter of the call or message and/or the time of day.

[0005] It is another object of the present invention to provide a call processing system which can first transcribe messages received by telephone, facsimile and e-mail, as well as other data electronically received by the system, then tag the identity of the caller (or participants to a teleconference) or the author of such e-mail or facsimile messages, and then index such calls, conversations and messages according to their origin and subject matter, whereby an authorized user can then access the system, either locally or remotely, to playback such telephone conversations or messages or retrieve such e-mail or facsimile messages in the form of synthesized speech.

[0006] It is yet another object of the present invention to provide a system that is responsive (i.e., accessible and programmable) to voice activated commands by an authorized user, wherein the system can identify and verify the user before allowing the user to access calls or messages or program the system.

[0007] In one aspect of the present invention, a programmable automatic call and message processing system comprises: server means for receiving an incoming call; speaker recognition means, operatively coupled to the server means, for identifying the caller; speech recognition means, operatively coupled to the server means, for determining subject matter and content of the call; switching means, responsive to the speaker recognition means and speech recognition means, for processing the call in accordance with the identity of the caller and/or the subject matter of the call; and programming means, operatively coupled to the server means, speaker recognition means, speech recognition means and the switching means for programming the system to perform the processing.

[0008] The system is preferably programmed by the user so as to process incoming telephone calls in a predetermined manner based on the identity of the caller. Such processing includes, but is not limited to, switching the call to another system, forwarding the call to another telecommunication terminal, directing the call to an answering machine to be recorded, placing the call on hold, or disconnecting the call.

[0009] In another aspect of the present invention, the system may be pre-programmed to process an incoming telephone call, facsimile or e-mail message according to their content, subject matter, or according to the time of the day they are received. Still further, the system may preferably be programmed to process an incoming telephone call, facsimile or e-mail message according to a combination of such factors, i.e., the identity of the caller, the subject matter and content of the call and the time of day. In addition, e-mail messages (and other messages created by application specific software such as **LOTUS NOTES**) may be processed in accordance with mood stamps, i.e., informational fields provided by certain mailing programs such as **LOTUS NOTES** which allow the sender to indicate the nature of the message such as the confidentiality or urgency of the message. For future e-mail or data exchange techniques, such information can be included in a header of the e-mail or facsimile. Further, the system may be programmed to prompt the caller to explicitly advise the system of the nature of the message. Still further, the system may be configured to retrieve and process data from other telecommunication devices such as voice mail systems or answering

machines.

[0010] In still a further aspect of the present invention, the call processing system of the present invention is capable of tagging the identity of a caller or the participants to a teleconference, while transcribing the message or conversations of such callers and participants. Consequently, the system can automatically manage telephone messages and conversations, as well as voice mail, e-mail and facsimile messages, by storing such calls and messages according to their subject matter or the identity of the caller or author, or both. Specifically, the present invention can, in combination with such identification and transcription, automatically index or prioritize the received telephone calls and e-mail and facsimile messages according to their origin and/or subject matter which allows an authorized user to retrieve specific messages, e.g., those messages that originated from a specific source or those which deal with similar or specific subject matter.

[0011] In another aspect of the present invention, the system includes text-to- speech capabilities which allows the system to prompt (i.e., query) the user or caller in the form of synthesized speech, to provide answers to questions or requests by the user or caller in synthesized speech and to playback e-mail and facsimile messages in synthesized speech. The system also includes playback capabilities so as to playback recorded telephone messages and other recorded audio data.

[0012] These and other objects, features and advantages of the present invention will become apparent from the following detailed description of illustrative embodiments thereof, which is to be read in connection with the accompanying drawings.

Fig. 1 is a block diagram illustrating general functions of an automatic call and data transfer processing system in accordance with the present invention;

Fig. 2 is a block diagram, as well as a flow diagram, illustrating the functional interconnection between modules for a call and data transfer processing system in accordance with an embodiment of the present invention; and

Figs. 3a and 3b are flow diagrams illustrating a method for call or data transfer processing in accordance with the present invention.

[0013] Referring to Fig. 1, a block diagram illustrating general functions of an automatic call and data transfer processing system of the present invention is shown. The present invention is an automatic call and data transfer processing machine that can be programmed by an authorized user (block 12) to process incoming telephone calls in a manner pre-determined by such user Although the present invention may be employed to process any voice data that may be received through digital or analog channels, as well as data received electronically and otherwise convertible into readable text (to be further explained below), one embodiment of the present invention involves the processing of telephone communications. Particularly, the system 10 will automatically answer an incoming telephone call from a caller (block 14) and, depending upon the manner in which the system 10 is programmed by the user (block 12), the system 10 may process the telephone call by, for example, switching the call to another telecommunication system or to an answering machine (Block 18), or by handling the call directly, e.g., by connecting, disconnecting or placing the caller on hold (Block 16). In addition, the system 10 may be programmed to route an incoming telephone call to various telecommunication systems in a specific order (e.g., directing the call to several pre-determined telephone numbers until such call is answered) or simultaneously to all such systems. It is to be understood that the telecommunication systems listed in block 18, as well as the options shown in block 16 of Fig. 1, are merely illustrative, and not exhaustive, of the processing procedures that the system 10 may be programmed to perform.

[0014] In another embodiment of the present invention, the system 10 may be programmed to process incoming facsimile and e-mail messages, or automatically retrieve messages from e-mail or voice mail systems. Thus, it is to be understood that the bidirectional lines of Fig. 1 connecting the system 10 to the telecommunication systems in block 18 (e.g., e-mail, voice mail, facsimile/modem and answering machine) indicates that the system 10 is designed to send data (e.g., calls or messages) to such systems, as well as retrieve and process data stored or recorded in such systems. For instance, the system 10 may be programmed to process a particular call by directing the call to an answering machine (block 18) to be recorded. The system 10 may subsequently retrieve the recorded message from the answering machine, which is then decoded and processed by the system 10 in a particular manner. Further, the system 10 can be programmed to transform an incoming telephone call or messages into a page which can then be transmitted to the user's pager, cellular phone or e-mail.

[0015] The functional modules of the system 10 and their specific interaction in accordance with an embodiment of the present invention will be explained below by reference to Fig. 2. It is to be understood that same or similar components illustrated throughout the figures are designated with the same reference numeral. It is to be further understood that the functional modules described herein in accordance with the present invention may be implemented in hardware, software, or a combination thereof. Preferably, the main speech and speaker recognition, language identification modules and indexing modules of present invention, for example, are implemented in software on one or more appropriately programmed general purpose digital computer or computers, each having a processor, associated mem-

ory and input/output interfaces for executing the elements of the present invention. It should be understood that while the invention is preferably implemented on a suitably programmed general purpose computer or computers, the functional elements of Fig. 2 may be considered to include a suitable and preferred processor architecture for practicing the invention and are exemplary of functional elements which may be implemented within such computer or computers through programming. Further, the functional elements of Fig. 2 may be implemented by programming one or more general purpose microprocessors. Of course, special purpose microprocessors may be employed to implement the invention. Given the teachings of the invention provided herein, one of ordinary skill in the related art will be able to contemplate these and similar implementations of the elements of the invention.

[0016] Referring now to Fig. 2, the system 10 includes a server 20 preferably connected to various telecommunication systems including, but not limited to, one or more telephone lines (block 14) and one or more facsimile and a modem lines (Figs. 1 and 2, block 18) for receiving and sending telephone calls and message data, respectively. The server 20 is programmed to automatically answer incoming telephone calls and receive incoming facsimile transmissions. The system 10 may also include a permanent internet/intranet connection for accessing a local network mail server, whereby the server 20 can be programmed to periodically connect to such local network mail server (via TCP/IP) to receive and process incoming e-mails, as well as send e-mail messages. Alternatively, if the system 10 is not permanently connected to a local network server, the system server 20 may be programmed to periodically dial an access number to an internet provider to retrieve or send e-mail messages. Such procedures may also be performed at the option of the user (as opposed to automatically monitoring such e-mail accounts) when the user accesses the system 10.

[0017] Further, as shown in Figs. 1 and 2 (block 18), the server 20 may be directly connected to voice mail systems and answering machines so as to allow the user to retrieve and process messages that have been recorded on such voice-mail and answering machine systems. If the system 10 is connected to a local network system, the server 20 may be programmed to periodically retrieve messages from other voice mail systems or answering machines which are not directly connected to the server 20, but otherwise accessible through the local network, so that the system 10 can then automatically monitor and retrieve messages from such voice mail systems or answering machines.

[0018] The server 20 includes a recorder 40 for recording and storing audio data (e.g., incoming telephone calls or messages retrieved from voice mail or answering machines), preferably in digital form. Furthermore, the server 20 preferably includes a compression/decompression module 42 for compressing the digitized audio data, as well as message data received via e-mail and facsimile, so as to increase the data storage capability of a memory (not shown) of the system 10 and for decompressing such data before reconstruction when such data is retrieved from memory.

[0019] A speaker recognizer module 22 and an automatic speech recognizer/natural language understanding (ASR/NLU) module 24 are operatively coupled to the server 20. The speaker recognizer module 22 determines the identity of the caller 14 and participants to a conference call from the voice data received by the server 20, as well as the author of a received facsimile or e-mail message. The ASR/NLU module 24 converts voice data and other message data received from the server 20 into readable text to determine the content and subject matter of such calls, conversations or messages. In addition, as further demonstrated below, the ASR/NLU module 24 processes verbal commands from an authorized user to remotely program the system 10, as well as to generate or retrieve messages. The ASR/NLU module 24 also processes voice data from callers and authorized users to perform interactive voice response (IVR) functions. A language identifier/translator module 26, operatively connected to the ASR/NLU module 24, is provided so that the system 10 can understand and properly respond to messages in foreign language when the system is used, for example, in a multi-language country such as Canada.

[0020] A switching module 28, operatively coupled to the speaker recognizer module 22 and the ASR/NLU module 24, processes data received by the speaker recognizer module 22 and/or the ASR/NLU module 24. The switching module performs a processing procedure with respect to incoming telephone calls or facsimile or e-mail messages (e.g., directing a call to voice-mail or answering machine) in accordance with a pre-programmed procedure.

[0021] An identification (ID) tagger module 30, operatively connected to the speaker recognizer module 22, is provided for electronically tagging the identity of the caller to the caller's message or conversation or tagging the identity of the author of an e-mail or facsimile message. Further, when operating in the background of a teleconference, the ID tagger 30 will tag the identity of the person currently speaking. A transcriber module 32, operatively connected to the ASR/NLU module 24, is provided for transcribing the telephone message or conversation, teleconference and/or facsimile message. In addition, the transcriber module 32 can transcribe a verbal message dictated by the user, which can subsequently be sent by the system 10 to another person via telephone, facsimile or e-mail.

[0022] An audio indexer/prioritizer module 34 is operatively connected to the ID tagger module 30 and the transcriber module 32. The audio indexer/prioritizer module 34 stores the transcription data and caller identification data which is processed by the transcriber module 32 and the ID tagger module 30, respectively,

as well as the time of the call, the originating phone number (via automatic number identification (ANI) if available) and e-mail address, in a pre-programmed manner, so as to allow the user to retrieve specific calls or messages from a particular party or those calls or messages which pertain to specific subject matter. Further, the audio indexer/prioritizer can be programmed to prioritize certain calls or messages and inform the user of such calls or messages.

[0023] A speech synthesizer module 36, operatively connected to the audio indexer/prioritizer module 34, allows the user to retrieve messages (e-mails or facsimiles) in audio form (i.e., synthesized speech). The speech synthesizer is also operatively coupled to the ASR/NLU module for providing system prompts (i.e., queries) in the form of synthesized speech (as opposed to being displayed, for example, on a computer monitor).

[0024] A programming interface 38, operatively coupled to the server 20, speaker recognizer module 22, language identifier/translator module 26, ASR/NLU module 24, audio indexer/prioritizer module 34 and the switching module 28, is provided for programming the system 10 to process calls and messages in accordance with a pre-determined procedure. As explained in detail below, a user may program the system 10 using the programming interface 38 through either voice commands or a GUI (graphical user interface), or both. In a preferred embodiment, the system 10 is programmed by verbal commands from the user (i.e., voice command mode). Specifically, the user may program the system 10 with verbal commands either remotely, by calling into the system 10, or locally with a microphone. The programming interface 38 is connected to the server 20 which, in conjunction with the speaker recognizer module 22 and the ASR/NLU module 24, verifies the identity of the user before processing the verbal programming commands of the user. The system 10 may either display (via the GUI) or play back (via the speech synthesizer 36) information relating to the verbal programming commands (i.e., whether the system 10 recognizes such command), as well as the current programming structure of the system 10.

[0025] In another embodiment, the system 10 may be programmed locally, through a PC and GUI screen or programmed remotely, by accessing the system 10 through a computer network from a remote location. Similar to conventional windows interface, the user may program the system 10 by selecting certain fields which may be displayed on the GUI. It is to be appreciated that the system 10 may be programmed through a combination of voice commands and a GUI. In such a situation, the GUI may, for example, provide assistance to the user in giving the requisite voice commands to program the system 10. Still further, the system 10 may be programmed by editing a corresponding programming configuration file which controls the functional modules of Fig. 2.

[0026] The operation of the present invention will now be described with reference to Fig. 2 and Figs. 3a and 3b. It is to be understood that the depiction of the present invention in Fig. 2 could be considered a flow chart for illustrating operations of the present invention, as well as a block diagram showing an embodiment of the present invention. The server 20 is programmed to automatically answer an incoming telephone call, e-mail, facsimile/modem, or other electronic voice or message data (step 100). The server 20 distinguishes between incoming telephone calls, e-mail messages, facsimile messages, etc., by special codes, i.e. protocols, at the beginning of each message which indicates the source. Particularly, the server 20 initially assumes that the incoming call is a telephone communication and will proceed accordingly (step 110) unless the server 20 receives, for example, a modem handshake signal, whereby the system 10 will handle the call as a computer connection protocol. It is to be understood that the system 10 may be programmed to monitor other voice mail or e-mail accounts by periodically calling and retrieving voice mail and e-mail messages from such accounts.

[0027] If it is determined that the incoming call received by the server 20 is a telephone call, the audio data (e.g., incoming calls as well as calls retrieved from voice mail or answering machines) is recorded by the recorder 40 (step 112). The recorder 40 may be any conventional device such as an analog recorder or digital audio tape ("DAT"). Preferably, the recorder 40 is a digital recorder, i.e., an analog-to-digital converter for converting the audio data into digital data. The digitized audio data may then be compressed by the compression/decompression module 42 (step 114) before being stored (step 116) in memory (not shown in Fig. 2). It is to be appreciated that any conventional algorithm, such as those disclosed in "Digital Signal Processing, Synthesis and Recognition" by S. Furui, Dekker, 1989, may be employed by the compression/decompression module 42 to process the message data.

[0028] Next, simultaneously with the recording and storing of the audio data, the identity of the caller is determined by processing the caller's audio communications and/or audio responses to queries by the system 10. Specifically, the caller's verbal statements and responses are received by the server 20 and sent to speaker recognizer module 22, wherein such verbal statements and responses are processed and compared with previously stored speaker models (step 120). If the speaker is identified by matching the received voice data with a previously stored voice model of such speaker (step 130), and if the system 10 is pre-programmed to process calls based on the identity of a caller, the system 10 will then process the telephone call in accordance with such pre-programmed procedure (step 152).

[0029] If, on the other hand, the speaker (e.g., a first time caller) cannot be identified via the previously

stored voice models, speaker identification may be performed by both the speaker recognizer module 22 and the ASR/NLU module 26, whereby the content of the telephone message may be processed by the ASR/NLU module 26 to extract the caller's name which is then compared with previously stored names to determine the identity of such caller (step 140). If the identity of the caller is then determined, the system 10 will process the telephone call in accordance with the identity of the caller (step 152).

[0030] In the event that the system 10 is unable to identify the caller from either the stored voice models or the content of the telephone message, the speaker recognizer module 22 sends a signal to the server 20 which, in turn, prompts the caller to identify him or herself with a query, e.g., "Who are you," (step 150) and the above identification process is repeated (step 120). The server 20 obtains the query in synthesized speech from speech synthesizer module 36 It is to be understood that, as stated above, the system 10 may be programmed to initially prompt the caller to identify him or herself or ask details regarding the reason for the call.

[0031] Once the caller or author has been identified by the speaker recognizer module 22, a signal is sent by the speaker recognizer module 22 to the switching module 28, whereby the switching module 28 processes the call or message based on the identity of the caller or author in accordance with a pre-programmed procedure (step 152). If, on the other hand, the identity of the caller ultimately cannot be identified, the system 10 may be programmed to process the call based on an unknown caller (step 154) by, e.g., forwarding the call to a voice mail. Such programming, to be further explained, is performed by the user 12 through the programming interface module 38. As stated above, the processing options which the system 10 may be programmed to perform include, but are not limited to, switching the call to another system, directing the call to another telecommunication terminal (Figs. 1 and 2, block 18) or directly handling the call by either connecting the call to a particular party, disconnecting the call, or placing the call on hold (Figs. 1 and 2, block 16).

[0032] It is to be appreciated that whenever a new caller interacts with the system 10 for the first time, speaker models are built and stored in the speaker recognizer module 22, unless erased at the option of the user. Such models are then utilized by the speaker recognizer module 22 for identification and verification purposes when that caller interacts with the system 10 at a subsequent time.

[0033] It is to be appreciated that the system 10 may perform speaker identification by utilizing methods other than acoustic features when the requisite voice models do not exist. For example, with regard to telephone calls, the system 10 may utilize additional information (e.g. caller ID) to enhance the accuracy of the system 10 and/or to identify first time callers.

[0034] As further explained below, the system 10 may be programmed to store the name and originating telephone number of every caller (or specified callers). Such capability allows the user to automatically send reply messages to callers, as well as dynamically create an address book (which is stored in the system 10) which can be subsequently accessed by the user to send a message to a particular person.

[0035] It is to be understood that depending upon the application, it is not necessary that the system 10 perform speaker recognition and natural language understanding in real time (i.e., simultaneously with the recording and during the time period of the actual telephone call) in every instance. For example, the system 10 can be programmed to query the caller (via IVR programming) to obtain relevant information (i.e., name and reason for call) at the inception of the call and store such information. The identification process may then be performed by the speaker recognizer module 22 or the ASR/NLU module 24 subsequent to the call by retrieving the stored audio data from memory (step 118) (as indicated by the dotted line in Fig. 3a)

[0036] It is to be understood that any type of speaker recognition system may be utilized by the speaker recognizer module 22 for identifying the caller. Preferably, the speaker recognition system employed in accordance with the present invention is the system which performs text-independent speaker verification and asks random questions, i.e., a combination of speech recognition, text independent speaker recognition and natural language understanding as disclosed in U.S. Serial No. 08/871,784, filed on June 11, 1997, and entitled: "Apparatus And Methods For Speaker Verification / Identification / Classification Employing Non-Acoustic And/Or Acoustic Models and Databases," the disclosure of which is incorporated herein by reference. More particularly, the text-independent speaker verification system is preferably based on a frame-by frame feature classification as disclosed in detail in U.S. Serial No. 08/788,471 filed on January 28, 1997 and entitled: "Text Independent Speaker Recognition for Transparent Command Ambiguity Resolution And Continuous Access Control," the disclosure of which is also incorporated herein by reference.

[0037] As explained in the above-incorporated reference U.S. Serial No. 08/871,784, text-independent speaker recognition is preferred over text-dependant or text-prompted speaker recognition because text independence allows the speaker recognition function to be carried out in parallel with other speech recognition-based functions in a manner transparent to the caller without requiring interruption for new commands or identification of a new caller whenever a new caller is encountered.

[0038] Next, referring to Fig. 3b (and assuming the system 10 is programmed to process calls based on the identity of a caller or author), if it is determined that the incoming call is a facsimile or e-mail message, the message data (e.g., incoming e-mails or messages

retrieved from e-mail accounts) are processed by the ASR/NLU module 24 (step 190), compressed (step 192), and stored (step 194) in memory (not shown). With regard to e-mail messages, the data is directly processed (since such data is already in text format). With regard to facsimile messages, the ASR/NLU module 24 employs optical character recognition (OCR) using known techniques to convert the received facsimile message into readable text (i.e., transcribe the facsimile message into an ASCII file).

[0039] Next, simultaneously with the transcribing and storing of the incoming message data, the identity of the author of such message may be determined via the ASR/NLU module 24 whereby the content of the incoming message is analyzed (step 200) to extract the author's name or the source of the message, which is then compared with previously stored names to determine the identity of such author (step 210). If the author is identified (step 210), the message can be processed in accordance with a pre-programmed procedure based on the identity of the author (step 222). If, on the other hand, the identity of the author cannot be identified, the message may be processed in accordance with the pre-programmed procedure for an unidentified author (step 224).

[0040] As stated above, it is to be understood that it is not necessary that the system 10 process the incoming or retrieved message in real time (i.e., simultaneously with the transcribing of the actual message) in every instance. Processing may be performed by the ASR/NLU module 24 subsequent to receiving the e-mail or facsimile message data by retrieving the transcribed message data from memory (step 196) (as indicated by the dotted line in Fig. 3b).

[0041] In addition to the identity of the caller or author, the system 10 may be further programmed by the user 12 to process an incoming telephone call or facsimile or e-mail message based on the content and subject matter of the call or message and/or the time of day in which such call or message is received. Referring again to Figs. 2, 3a and 3b, alter receiving an incoming telephone call or e-mail or facsimile message, or alter retrieving a recorded message from an answering machine or voice mail, the server 20 sends the call or message data to the ASR/NLU module 24. In the case of voice data (e.g. telephone calls or messages retrieved from voice mail or answering machine), the ASR/NLU module 24 converts such data into symbolic language or readable text. As stated above, e-mail messages are directly processed (since they are in readable text format) and facsimile messages are converted into readable text (i.e., ASCII files) via the ASR/NLU module 26 using known optical character recognition (OCR) methods. The ASR/NLU module 26 then analyzes the call or message data by utilizing a combination of speech recognition to extract certain keyword or topics and natural language understanding to determine the subject matter and content of the call (step 160 in Fig. 3a for telephone calls) or message (step 200 in Fig. 3b for e-mails and facsimiles).

[0042] Once the ASR/NLU module determines the subject matter of the call (step 170 in Fig. 3a) or the message (step 220 in Fig. 3b), a signal is then sent to the switching module 28 from the ASR/NLU module 24, wherein the call or message is processed in accordance with a pre-determined manner based on the subject matter and content of the call (step 158 in Fig. 3a) or the content of the message (step 228 in Fig. 3b). For instance, if a message or call relates to an emergency or accident, the switching module 28 may be programmed to transfer the call immediately to a certain individual.

[0043] In the event that the ASR/NLU module 24 is unable to determine the subject matter or content of a telephone call, the ASR/NLU module 24 sends a signal to the speech synthesizer 36 which, in turn, sends a message to the server 20, to prompt the caller to articulate in a few words the reason for the call (step 180), e.g., "What is the reason for your call?" Again, it is to be understood that the system 10 may be programmed to initially prompt the caller to state the reason for the call. If the system 10 is still unable to determine the subject matter of such call, the call may be processed in accordance with a pre-programmed procedure based on unknown matter (step 156) Likewise, if the subject matter of an e-mail or facsimile message cannot be determined (step 220), the message may be processed in accordance with a pre-programmed procedure based on unknown matter (step 226).

[0044] Further, in the event that an incoming call or e-mail message is in a language foreign to the system 10 (i.e., foreign to the user), the ASR/NLU module 26 will signal the language identifier/translator module 26 to identify the particular language of the call or message, and then provide the required translation to the ASR/NLU module 26 so as to allow the system 10 to understand the call and answer the caller in the proper language. It is to be understood that the system 10 may also be pre-programmed to process calls or messages with an unknown language in a particular manner.

[0045] It is to be appreciated that any conventional technique for language identification and translation may be employed in the present invention, such as the well-known machine language identification technique disclosed in the article by Hieronymus J. and Kadambe S., "Robust Spoken Language Identification using Large Vocabulary Speech Recognition," Proceedings of ICASSP 97, Vol. 2 pp. 1111, as well as the language translation technique disclosed in Hutchins and Somers (1992): "An Introduction to Machine Translation," Academic Press, London; (encyclopedic overview).

[0046] In addition to the above references, language identification can be performed using several statistical methods. First, if the system 10 is configured to process a small number of different languages (e.g., in Canada where essentially only English or French are spoken),

the system 10 may decode the input text in each of the different languages (using different ASR systems). The several decoded scripts are then analyzed to find statistical patterns (i.e., the statistical distribution of decoded words in each script is analyzed). If the decoding was performed in the wrong language, the perplexity of the decoded script would be very high, and that particular language would be excluded from consideration.

[0047] Next, language identification may be performed on a phonetic level where the system recognizes a set of phonemes (either using a universal phonetic system or several systems for different languages). The system then estimates the frequencies of the decoded phoneme sequences for each language. If a particular decoded sequence is unusual, the system would exclude such language from consideration. There may also be some sequences which are typical for a certain language. Using such factors, the system will identify the most probable language.

[0048] It is to appreciated that the present invention may utilize the identity of the caller to perform language identification. Specifically, if the speaker profile of a certain caller (which is stored in the system 10) indicates that the caller speaks in a certain language, this information may be a factor in identifying the language. Conversely, if the system 10 identifies a particular language using any of the above methods, the system 10 may then determine the identity of a caller by searching the speaker profiles to determine which speakers use such identified language.

[0049] It is to be understood that both speech recognition and natural language understanding may be utilized by the ASR/NLU module 24 to process data received from the server 20. The present invention preferably employs the natural language understanding techniques disclosed in U.S. Serial No. 08/859,586, filed on May 20, 1997, and entitled: "A Statistical Translation System with Features Based on Phrases or Groups of Words," and U.S. Serial No. 08/593,032, filed on January 29, 1996 and entitled "Statistical Natural Language Understanding Using Hidden Clumpings," the disclosures of which are incorporated herein by reference. The above-incorporated inventions concern natural language understanding techniques for parameterizing (i.e. converting) text input (using certain algorithms) into language which can be understood and processed by the system 10. For example, in the context of the present invention, the ASR component of the ASR/NLU module 24 supplies the NLU component of such module with unrestricted text input such as "Play the first message from Bob." Such text may be converted by the NLU component of the ASR/NLU module 24 into "retrieve-message(sender=Bob, message-number=1)." Such parameterized action can then be understood and acted upon by the system 10.

[0050] The known automatic speech recognition functions disclosed in the article by Zeppenfeld, et al., entitled "Recognition of Conversational Telephone Speech Using The Janus Speech Engine," Proceedings of ICASSP 97, Vol. 3, pp. 1815 1997; and the known natural language understanding functions disclosed in the article by K. Shirai and S. Furui, entitled "Special Issue on Spoken Dialog," 15, (3-4) Speech Communication, 1994 may also be employed in the present invention. Further, to simplify the programming of the ASR/NLU module 24, the keyword spotting based recognition methods as disclosed in "Word Spotting from Continuous Speech Utterances," Richard C. Cross, Automatic Speech and Speaker Recognition, Advanced Topics, pp. 303-327, edited by Chin-Hui Lee, Frank K. Soong, Kuldip K. Paiwal (Huwer Academic Publishers), 1996 may preferably be used to guarantee that certain critical messages are sufficiently handled.

[0051] It is to be appreciated that by utilizing natural language understanding, as demonstrated above, the system 10 is capable of performing interactive voice response (IVR) functions so as to establish a dialog with the user or caller to provide dialog management and request understanding. This enables the system 10 to be utilized for order taking and dialog-based form filing. Further, such functions allow the caller to decide how to process the call (assuming the system 10 is programmed accordingly), i.e., by leaving an e-mail or voice mail message, sending a page or transferring the call to another telephone number. In addition, to be explained below, this allows the system 10 to be remotely programmed by the user through voice commands.

[0052] It is to be further appreciated that the system 10 provides security against unauthorized access to the system 10. Particularly, in order for a user to have access to and participate in the system 10, the user must go through the system's enrollment process. This process may be effected in various ways. For instance, enrollment may be performed remotely by having a new user call and enter a previously issued personal identification number (PIN), whereby the server 20 can be programmed to respond to the PIN which is input into the system 10 via DTMF Keys on the new user's telephone. The system 10 can then build voice models of the new user to verify and identify the new user when he or she attempts to access or program the system 10 at a subsequent time. Alternatively, either a recorded or live telephone conversation of the new user may be utilized to build the requisite speaker models for future identification and verification.

[0053] It is to be appreciated that the server 20 of the present invention may be structured in accordance with the teachings of patent application (IBM Docket Number Y0997-313) entitled "Apparatus and Methods For Providing Repetitive Enrollment in a Plurality of Biometric Recognition Systems Based on an Initial Enrollment," the disclosure of which is incorporated by reference herein, so as to make the speaker models (i.e., biometric data) of authorized users (which are stored in the server 20) available to other biometric recognition

based systems to automatically enroll the user without the user having to systematically provide new biometric models to enroll in such systems.

[0054] The process of programming the system 10 can be performed by a user either locally, via a GUI interface or voice commands, or remotely, over a telephone line (voice commands) or through a network system connected to the system. In either event, this is accomplished through the programming interface 38. As demonstrated above, programming the system 10 is achieved by, e.g., selecting the names of persons who should be transferred to a certain number, voice mail or answering machine, by inputting certain keywords or topics to be recognized by the system 10 as requiring certain processing procedures and/or by programming the system 10 to immediately connect emergency calls or business calls between the hours of 8:00 a.m. and 12:00 p.m. As shown in Fig. 2, the programming interface 38 sends such information to the server 20, speaker recognizer module 22, ASR/NLU module 26, language identifier/translator module 24, audio indexer/prioritizer module 34 and the switching module 28, which directs the system 10 to process calls in accordance with the user's programmed instructions.

[0055] The programming interface is responsive to either DTMF key signal or voice commands by an authorized user. The preferred method of programming the system 10 is through voice activated commands via a process of speech recognition and natural language understanding, as opposed to DTMF keying or via GUI interface. This process allows the system 10 to verify and identify the user before the user is provided access to the system 10. This provides security against unauthorized users who may have knowledge of an otherwise valid PIN. Specifically, before the user can program the system 10 through voice commands, the user's voice is first received by server 20, and then identified and verified by the speaker recognizer module 22. Once the user's identification is verified, the server 20 will signal the programming interface 38 to allow the user to proceed with programming the system 10.

[0056] The voice commands for programming the system 10 are processed in the ASR/NLU module 24. Particularly, during such programming, the ASR/NLU module 24 is in a command and control mode, whereby every voice instruction or command received by the programming interface 38 is sent to the ASR/NLU module 24, converted into symbolic language and interpreted as a command. For instance, if the user wants the system 10 to direct all calls from his wife to his telephone line, the user may state, e.g., "Immediately connect all calls from my wife Jane," and the system 10 will recognize and process such programming command accordingly.

[0057] Moreover, the user can establish a dialog with the system 10 through the ASR/NLU module 24 and the speech synthesizer module 35. The user can check the current program by asking the programming interface 38, e.g., "What calls are transferred to my answering machine." This query is then sent from the server 20 (if the user is calling into the system 10 from an outside line), or from the programming interface 28 via the server 20 (if the user is in the office), to the ASR/NLU module 24, wherein the query is processed. The ASR/NLU 24 module will then generate a reply to the query, which is sent to the speech synthesizer 36 to generate a synthesized message, e.g., "All personal calls are directed to your answering machine," which is then played to the user.

[0058] Similarly, if the system 10 is unable to understand a verbal programming request from an authorized user, the ASR/NLU module 24 can generate a prompt for the user, e.g., "Please rephrase your request," and processed by the speech synthesizer 36. Specifically, during such programming, the server 20 sends a programming request to the programming interface 38. If the system 10 is unable to decipher the request, the programming interface 38 sends a failure message back to the server 20, which relays this message to the ASR/NLU module 24. The ASR/NLU module 24 may then either reprocess the query for a potential different meaning, or it can prompt the user (via the speech synthesizer 36) to issue a new programming request.

[0059] It is to be appreciated that the system 10 may be programmed to manage various messages and calls received via voice-mails, telephone lines, facsimile/modem, e-mail and other telecommunication devices which are connected to the system 10 through the operation of the audio indexer/prioritizer module 34. In particular, the audio indexer/prioritizer module 34 may be programmed to automatically sort and index such messages and telephone conversations according their subject matter and content, origin, or both. The system 10 can preferably be further programmed so as to prioritize certain calls and messages from a specific individual.

[0060] Referring to Fig. 2, the audio indexing feature of the system 10 works as follows. Once the caller is identified and verified by the speaker recognizer module 22, the speaker recognizer module 22 signals the ID tagger module 30 which automatically tags the identity of the caller or the identity of current speaker of a group of participants to a teleconference. Simultaneously with the ID tagging process, the transcriber module 32 transcribes the telephone conversation or message. The tagging process involves associating the transcribed message with the identity of the caller or speaker. For instance, during teleconferences, each segment of the transcribed conversation corresponding to the current speaker is tagged with the identity of such speaker together with the begin time and end time for each such segment.

[0061] The information processed in the ID tagger module 30 and the transcriber module 32 is sent to the audio indexer/prioritizer module 34, wherein the received information is processed and stored according

to a pre-programmed procedure. The audio indexer/prioritizer module 34 can be programmed to index the messages and conversations in any manner that the user desires. For instance, the user may be able to either retrieve the messages from a certain caller, retrieve all urgent messages, or retrieve the messages that relate to a specific matter. Further, the audio indexer/prioritizer module 34 can be programmed to prioritize calls from a caller who has either left numerous messages or has left urgent messages.

[0062] The information stored in the audio indexer/prioritizer module 36 can then be accessed and retrieved by the user either locally or remotely. When such information is accessed by the user, the audio indexer/prioritizer module 36 send the requested information to the speech synthesizer module 38, wherein a text-to-speech conversion is performed to allow the user to hear the message in the form of synthesized speech. It is to be understood that any conventional speech synthesizing technique may be utilized in the present invention such as the Eloquent engine provided with the commercially available IBM **VIAVOICEGOLD** software.

[0063] It is to be appreciated that information may be retrieved from the audio indexer/prioritizer module 34 through various methods such as via GUI interface, PINs and DTMF keying. The preferred method in the present invention for retrieving such information, however, is through voice activated commands. Such method allows the system 10 to identify and verify the user before providing access to the messages or conversations stored and indexed in the audio indexer/prioritizer module 34. The audio indexer/prioritizer module 34 can be programmed to recognize and respond to certain voice commands of the user, which are processed by the ASR/NLU module 24 and sent to the audio indexer/prioritizer module 34, in order to retrieve certain messages and conversations. For example, the user may retrieve all the messages from Mr. Smith that are stored in the audio indexer/prioritizer module 36 through a voice command, e.g., "Play all messages from Mr. Smith." This command is received by the server 20 and sent to the ASR/NLU module 24 for processing. If the ASR/NLU module 24 understands the query, the ASR/NLU MODULE 24 sends a reply back to the server 20 to process the query. The server 20 then signals the indexer/prioritizer module 34 to send the requested messages to the speech synthesizer to generate synthesized e-mail or facsimile messages, or directly to the server 20 for recorded telephone or voice mail messages, which are simply played back.

[0064] It is to be appreciated that various alternative programming strategies to process calls may be employed in the present invention by one of ordinary skill in the art. For instance, the system 10 may be programmed to warn the user in the event of an important or urgent incoming telephone call. Specifically, the system 10 can be programmed to notify the user on a display thereby allowing the user to make his own decision on how to handle such call, or to simply process the call, as demonstrated above, in accordance with a pre-programmed procedure. Moreover, the system 10 can be programmed to forward an urgent or important call to the user's beeper when the user is not home or is out of the office. The user may also program the system 10 to dial a sequence of telephone numbers (alter answering an incoming telephone call) at certain locations where the user may be found during the course of the day. Furthermore, the sequence (i.e., list) of pre-programmed telephone numbers may be automatically updated by the system 10 in accordance with the latest known location where the user is found. If the user desires, such list may also accessible by individuals who call into the system 10 so that such callers can attempt to contact the user at one of the various locations at their convenience.

[0065] In addition, it is to be appreciated that the system 10 may be programmed to store the names of all persons who call the system 10, together with their telephone numbers (using ANI), as well as e-mail addresses of persons who send electronic mail. This allows the user of the system 10 to automatically reply to pending calls or messages without having to first determine the telephone number or e-mail addresses of the person to whom the user is replying. Further, such programming provides for dynamically creating a continuously up-to-date address book which is accessible to an authorized user to send messages or make calls. Specifically, the user can access the system 10, select the name of a particular person to call, and then command the system 10 to send that person a certain message (e.g., e-mail or facsimile).

[0066] Furthermore, the system 10 may be programmed to allow the callers to access and utilize specific functions of the system 10. For instance, the system 10 may offer the caller the option to schedule a tentative appointment with the user, which may then be stored in the system 10 and then subsequently accepted or rejected by the user. The caller may also be afforded the opportunity to chose the method by which the user may confirm, reject or adjourn such appointment (e.g., telephone call, facsimile or e-mail). Additionally, the system 10 may be programmed to provide certain authorized caller with access to the user's appointment calendar so that such appointments may be easily scheduled.

[0067] It is to be further appreciated that the present invention may be employed in a small scale application for personal home use, or employed in a large scale office or corporate applications. It is to be further appreciated by one of ordinary skill in the art that the system 10 may be utilized in other applications. For instance, by utilizing the NLU feature of the system 10, the system 10 may be connected to devices such as tape recorders, radios and televisions so as to warn the user whenever a certain topic is being covered on some channel or if a particular person is being interviewed. It is to be

understood that the system 10 is not limited to telephone communications. It is possible to use the system 10 for web phones, net conversations, teleconferences and other various voice communications which involve the transmission of voice through a digital or analog channel. Additional electronic information such as ASCII characters, facsimile messages and the content of web pages and database searches can also be processed in the same manner. For example, by adding optical character recognition (OCR) with facsimile receiving capabilities, the system 10 is able to transcribe the content of messages received by facsimile or e-mail to be stored in the audio indexer/prioritizer 34. As demonstrated above, the user may then retrieve these messages through the speech synthesizer 36 to hear the content of such messages.

[0068] In sum, the present invention provides a programmable call and message processing system which can be programmed by a user to process incoming telephone calls, e-mails messages, facsimile messages and other electronic information data in a predetermined manner without the user having to first manually answer a telephone call or retrieve an e-mail or facsimile message, identify the caller or the author of the message, and then decide how to transfer such call or respond to such message. The present invention can be programmed to transcribe telephone conversations or teleconferences, tag the identity of the caller or participants to the teleconference, and store such messages and conversations according to the identity of the caller or author and/or the subject matter and content of the call or message. The user may then retrieve any stored message or conversation based on the identity of the caller or a group of related messages based on their subject matter.

Further features of the invention may be as follows:

[0069] The server means further receives, and is responsive to, one of an incoming facsimile message, e-mail message, voice data, data convertible to text and a combination thereof

[0070] The speaker recognition means is based on text-independent speaker recognition.

[0071] The speech recognition means utilizes speech recognition and natural language understanding to determine said subject matter and content of said call.

[0072] The system includes language identification means, operatively coupled to said speech recognition means, for identifying and understanding languages of said incoming call.

[0073] The identification means performs language translation.

[0074] The identity of said caller is determined from said identified language of said call.

[0075] The language identification means uses identity of said caller to identify language of said call.

[0076] Enrollment means and includes for enrolling a new user to have access to said system.

[0077] The new user may be self-enrolled.

[0078] Means are provided for determining a time of said call and wherein said system may be further programmed to process said call in accordance with said time of said call.

[0079] The programming means includes one of a GUI interface, a voice interface, a programming configuration file, and a combination thereof

[0080] The programming may be performed one of locally, remotely and a combination thereof.

[0081] Means are provided, responsive to said incoming call, for dynamically creating an address book.

[0082] Means are provided for accessing said address book to send a message to a selected person.

[0083] Processing of said call includes transferring an incoming telephone call to a plurality of different telephone numbers one of sequentially and simultaneously.

[0084] Means are provided for prompting the caller to identify him/herself and the subject matter of said call. Said prompting is performed when said system cannot determine either said identity or said subject matter of call.

[0085] Alternately said prompting is performed when said call is received to determine said identity of said caller and subject matter of said call.

[0086] May further comprise means, operatively connected to said transcribing means, for dictating messages from a user of said system and sending said message to a selected person. The message may be sent by one of a facsimile, e-mail or telephone call, and a combination thereof, to said selected person.

[0087] May further comprise means for adding mood stamps or urgency/confidentiality stamps in a header in one of said facsimile and e-mail.

[0088] The step of determining said identity of said caller may be performed by text-independent speaker recognition.

[0089] The step of determining said subject matter of said call may be performed by speech recognition and natural language understanding.

[0090] The method may include the step of translating said call into a language other than that of said call.

[0091] The incoming call may be recorded.

[0092] Recording is performed simultaneously with said step of determining identity of said caller and may be performed prior to said step determining identity of said caller.

[0093] May further comprising the steps of: determining a time of said call; and processing said call based on said determined time of said call.

[0094] The the step of retrieving said indexed information is performed by voice commands.

[0095] The method may include determining the time of one of said call and message; and processing one of said call and message in accordance with said determined time

Claims

1. An automatic call and data transfer processing system, comprising: server means (20) for receiving an incoming call; characterised by

   speaker recognition means (22), operatively coupled to said server means, for identifying caller of said call;
   speech recognition means (24), operatively coupled to said server means, for determining subject matter and content of said call;
   switching means (28), responsive to said speaker recognition means and speech recognition means, for processing said call in accordance with one of said identification of said caller and determined subject matter; and
   programming means (38), operatively coupled to said server means, said speaker recognition means, said speech recognition means and said switching means for programming system to perform said processing.

2. A system of claim 1, characterised in that the server means includes means for recording (40) said incoming call.

3. A system of claim 2, characterised in that said server means further includes means (42) for compressing and storing said recorded data and means for decompressing said compressed data.

4. A system of claim 1, 2 or 3 further characterised by identification tagging means (30), responsive to said speaker recognition means, for automatically tagging said identity of said caller; transcribing means (32), responsive to said speech recognition means, for transcribing a telephone conversation or message of said caller; and audio indexing means (34), operatively coupled to said identification tagging means and said transcribing means, for indexing said messages and said conversations of said caller according to subject matter of said conversation and said message and the identity of said caller.

5. A system of claim 4 further characterised by means for retrieving (118) said indexed messages from said audio indexing means.

6. A system of claim 2, 4 or 5, further characterised by speech synthesizer means (36) operatively coupled to said server means, said speech recognition means and said audio indexing means, for converting information stored in said audio indexing means into synthesized speech.

7. A method for providing automatic call or message data processing, characterised by determining the identity of said caller (130) from an incoming call; determining the subject matter of said call (170); processing (152, 154, 156, 158) said call in accordance with one of said identity of said caller and subject matter of said call.

8. A method for providing automatic call or message data processing, comprising the steps of: receiving one of an incoming call and message data (100); identifying a caller of said call if an incoming call is received (130) and determining subject matter of said call (160); identifying an author of said message if message data is received and determining subject matter of said message; processing (152, 154, 156, 158) one of said call and message in accordance with one of said identity of said caller and author and said subject matter of said call and message.

9. The method further characterised by the steps of: tagging said determined identity of one of said caller and said author; transcribing said determined subject matter of one of said call and said message; indexing the information resulting from said tagging and said transcribing in accordance with one of said determined subject matter, said determined identity and a combination thereof.

10. A method may of claim 9 characterised by retrieving said indexed information and converting said indexed information into synthesized speech.

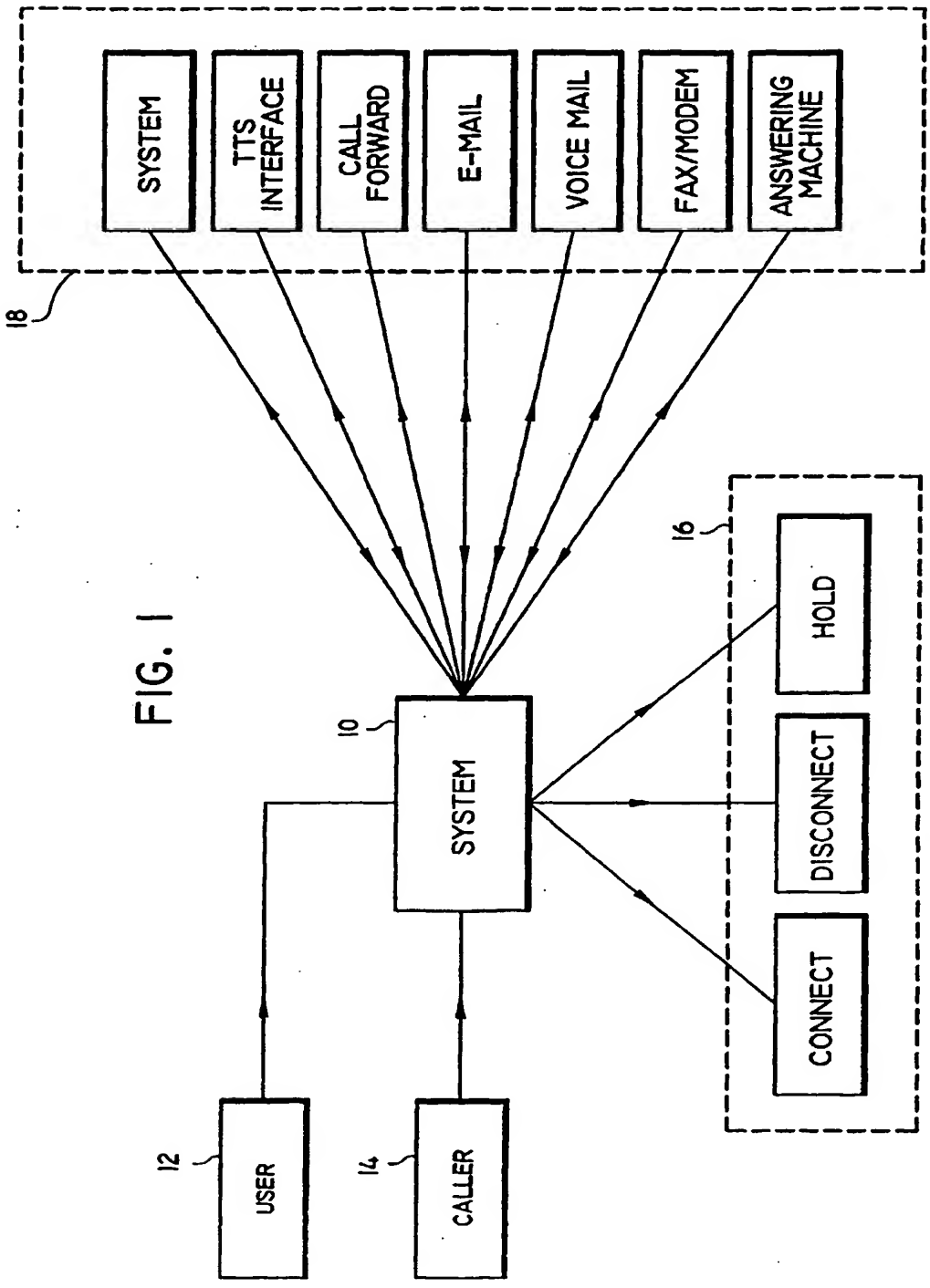

SYSTEM
TTS INTERFACE
CALL FORWARD
E-MAIL
VOICE MAIL
FAX/MODEM
ANSWERING MACHINE
18
16
HOLD
DISCONNECT
CONNECT
10
SYSTEM
12
USER
14
CALLER

FIG. 1

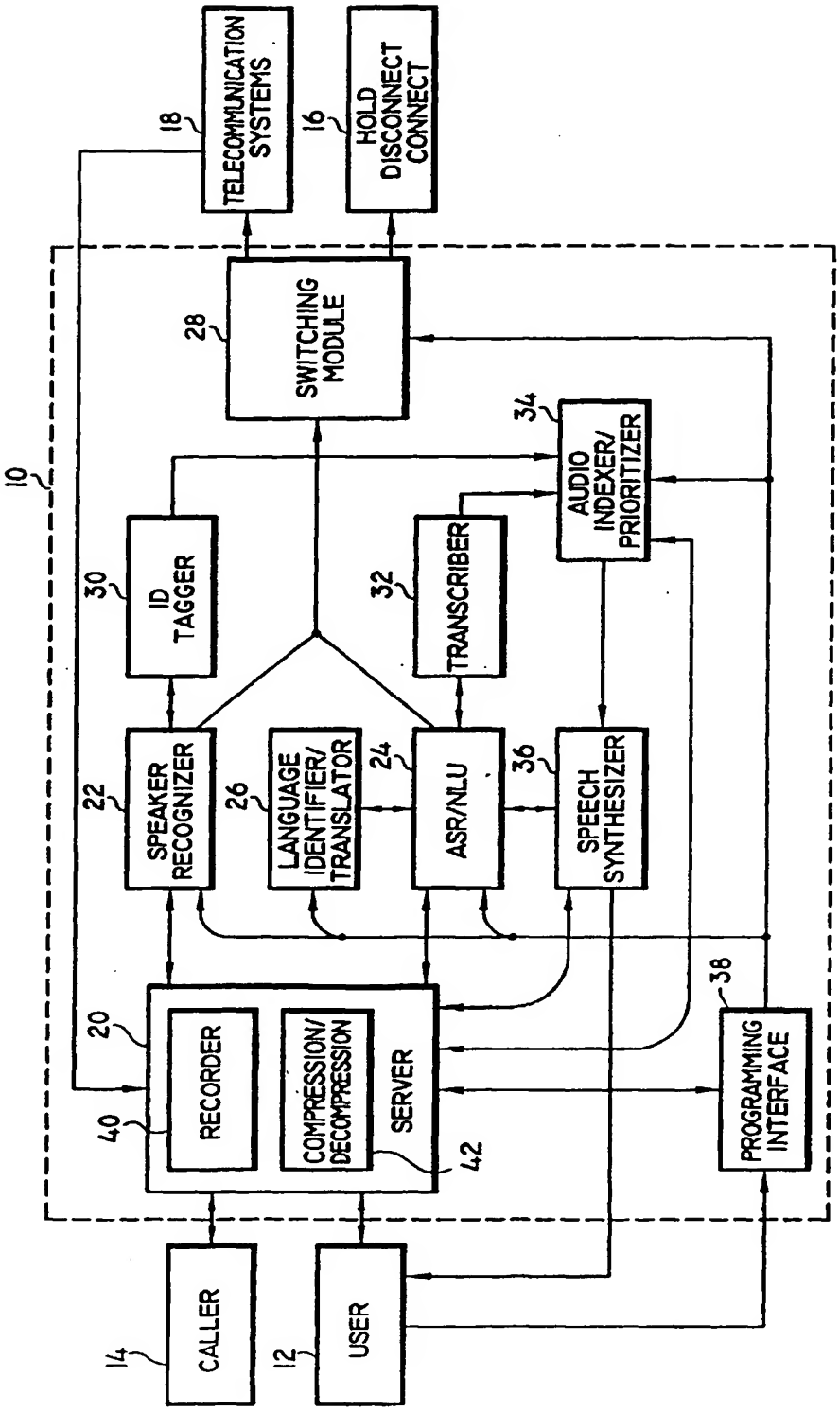
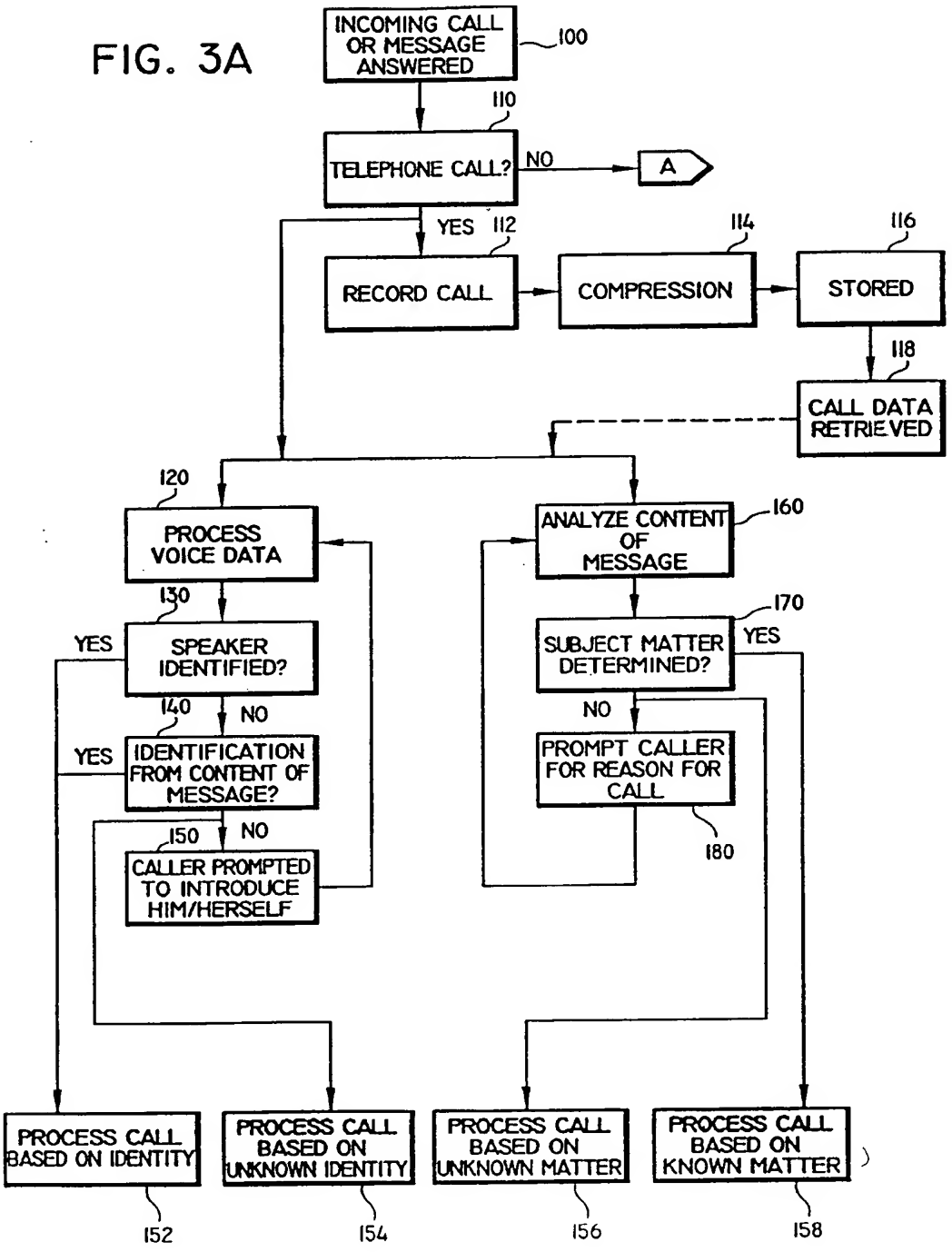

10
TELECOMMUNICATION SYSTEMS
18
HOLD DISCONNECT CONNECT
16
SWITCHING MODULE
28
AUDIO INDEXER/ PRIORITIZER
34
ID TAGGER
30
TRANSCRIBER
32
SPEAKER RECOGNIZER
22
LANGUAGE IDENTIFIER/ TRANSLATOR
26
ASR/NLU
24
SPEECH SYNTHESIZER
36
20
RECORDER
40
COMPRESSION/ DECOMPRESSION
SERVER
42
PROGRAMMING INTERFACE
38
CALLER
14
USER
12

FIG. 2

FIG. 3A


INCOMING CALL OR MESSAGE ANSWERED
100
110
TELEPHONE CALL?
NO
A
YES
112
RECORD CALL
114
COMPRESSION
116
STORED
118
CALL DATA RETRIEVED
120
PROCESS VOICE DATA
130
SPEAKER IDENTIFIED?
YES
140
NO
IDENTIFICATION FROM CONTENT OF MESSAGE?
YES
150
NO
CALLER PROMPTED TO INTRODUCE HIM/HERSELF
160
ANALYZE CONTENT OF MESSAGE
170
SUBJECT MATTER DETERMINED?
YES
NO
PROMPT CALLER FOR REASON FOR CALL
180
PROCESS CALL BASED ON IDENTITY
152
PROCESS CALL BASED ON UNKNOWN IDENTITY
154
PROCESS CALL BASED ON UNKNOWN MATTER
156
PROCESS CALL BASED ON KNOWN MATTER
158

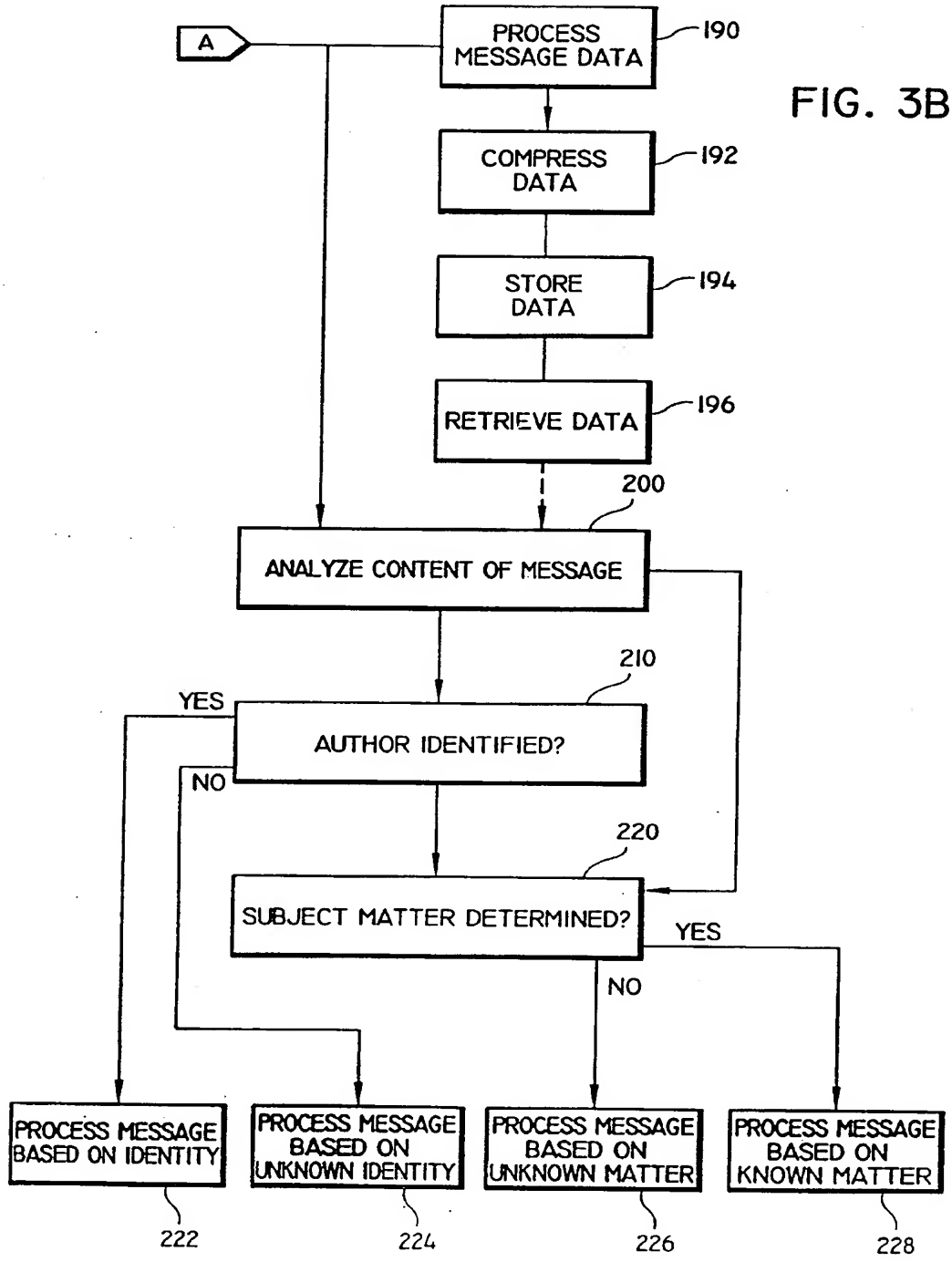

A
PROCESS MESSAGE DATA
190
COMPRESS DATA
192
STORE DATA
194
RETRIEVE DATA
196
200
ANALYZE CONTENT OF MESSAGE
210
YES
AUTHOR IDENTIFIED?
NO
220
SUBJECT MATTER DETERMINED?
YES
NO
PROCESS MESSAGE BASED ON IDENTITY
PROCESS MESSAGE BASED ON UNKNOWN IDENTITY
PROCESS MESSAGE BASED ON UNKNOWN MATTER
PROCESS MESSAGE BASED ON KNOWN MATTER
222
224
226
228

FIG. 3B